MG-SOFT Corporation

Visual YANG Designer 2015
Professional Edition

GETTING STARTED GUIDE

(Document Version: 4.3)

Document published on Friday, 16-January-2015

Copyright © 2010-2015 MG-SOFT Corporation

In order to improve the design or performance characteristics, MG-SOFT reserves the right to make changes in this document or in the software without notice.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of MG-SOFT Corporation. Permission to print one copy is hereby granted if your only means of access is electronic.

Depending on your license, certain functions described in this document may not be available in the version of the software that you are currently using.

Screenshots used in this document may slightly differ from those on your display.

MG-SOFT may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright © 2010-2015 MG-SOFT Corporation. All rights reserved.

TABLE OF CONTENTS

1 Introduction 7
1.1 Product Description 7
2 Install Visual YANG Designer Professional Edition 9
2.1 Requirements 9
2.1.1 Windows Operating System 9
2.1.2 Linux Operating System 9
2.1.3 Mac OS X Operating System 10
2.2 Installing Visual YANG Designer 10
2.2.1 Windows Operating System 10
2.2.2 Linux Operating System 11
2.2.3 Mac OS X Operating System 12
3 Start Visual YANG Designer Professional Edition 13
3.1 Starting Visual YANG Designer 13
3.1.1 Windows Operating System 13
3.1.2 Linux Operating System 13
3.1.3 Mac OS X Operating System 13
3.2 Visual YANG Designer Desktop 14
3.2.1 Two Modes of Operation: Design Mode and Source Mode 14
Design Mode 15
Source Mode 16
4 Apply License Key 19
5 Design New YANG Module in Design Mode 20
5.1 Creating a New Module 20
5.1.1 Adding Nodes to a New Module 25
Filtering Components in the Components Panel (Components Tab) 26
Building a YANG Schema Tree 28
5.2 Importing Modules 32
5.2.1 Using Nodes from Imported Modules 35
5.2.2 Copying Nodes from Imported Modules 38
5.3 Creating a Submodule 39
5.3.1 Adding Nodes to a Submodule 41
5.4 Saving Modules 42
6 Design New YANG Module in Source Mode 43
6.1 Switching to Source Mode 43
6.2 Creating a New Module in Source Mode 43
6.3 Using YANG Source Editor 44
6.3.1 Syntax Coloring and Pretty Print Formatting 44
6.3.2 Writing YANG Code and Using Auto-Complete Feature 45
6.3.3 Using Undo/Redo and Go to Last Edit Location Functions 46
6.3.4 Committing Changes 47
6.3.5 Saving Changes 47
6.3.6 Inserting Comments 49
6.3.7 Using Find Usages Feature 50
6.3.8 Using Go to Definition Feature 51
6.3.9 Using the Find/Replace Text Feature 53
7 Open Existing YANG Modules for Editing 56

7.1 Opening YANG and YIN Modules....56
7.2 Loading Known YANG and YIN Modules....60
8 Validate Modules....63
9 Viewing Resolved YANG Tree....65
10 Export Module to YANG (.yang) or YIN (.yin) Format....66
Appendix 1: List of Validation Messages....68
Appendix 2: List of Keyboard Shortcuts....71

TABLE OF FIGURES

Figure 1: Run dialog box 10
Figure 2: Visual YANG Designer desktop - Design mode 14
Figure 3: Selecting the desired mode of operation in Visual YANG Designer 15
Figure 4: Visual YANG Designer desktop - Source mode 17
Figure 5: Selecting and applying the license.key file 19
Figure 6: Specifying the basic parameters for a new YANG module 21
Figure 7: A newly added module node and its properties 23
Figure 8: Click a disabled input field of a module property to enable setting its value 24
Figure 9: Applying changes made in the Node Properties window panel 24
Figure 10: Adding an additional revision property to a module node 25
Figure 11: Example of configuration and state data nodes displayed in the YANG Tree panel 26
Figure 12: Example of a state data node with *Config* property set to “false” (empty *Config* checkbox) 26
Figure 13: The Components tab in the Components panel is context sensitive by default 27
Figure 14: Adding a *container* node to *module* node using drag&drop technique 28
Figure 15: A newly added container node and (some of) its default properties 29
Figure 16: Adding a *container* node to *module* node using the pop-up menu 29
Figure 17: A newly added leaf node and the error indication 31
Figure 18: The error message and icon disappears after setting the Type property of the leaf node 32
Figure 19: Importing a YANG module 32
Figure 20: Generate Import Statements dialog box 33
Figure 21: Imported YANG module displayed in the Components panel - Imports tab 34
Figure 22: Selecting the option to view the properties of a node in imported module 34
Figure 23: Viewing properties of the selected node as defined in imported module 35
Figure 24: Selecting a type defined in the imported module (e.g., inet:ipv4-address) 36
Figure 25: Adding a grouping node from the imported module (drag&drop) 37
Figure 26: A uses node with imported grouping 37
Figure 27: Copying a container node from the imported module (drag&drop+CTRL) 38
Figure 28: A copy of a subtree from imported module is added to the current module 39
Figure 29: Specifying the basic parameters for the new submodule 40
Figure 30: A newly added submodule node and its properties 40
Figure 31: Setting the revision property of a submodule node 41
Figure 32: A newly created *module* statement in the YANG Source Editor panel 44
Figure 33: the auto-complete drop-down list and description tooltip in the YANG Editor 45
Figure 34: Committing changes made in YANG Editor 47
Figure 35: Saving the changes 48
Figure 36: Commit and Save dialog box 48
Figure 37: Comments in YANG Source Editor 49
Figure 38: Selecting the *Find Usages* command 50
Figure 39: Inspecting the usage of the selected reusable code element 51
Figure 40: Selecting the *Go to Definition* command 52
Figure 41: Viewing the referenced reusable definition in YANG Source Editor 52
Figure 42: The Find/Replace toolbar in the Source mode 53
Figure 43: Finding and replacing text (matches are highlighted and displayed in annotation stripe) 54

Figure 44: Selecting commands for opening existing files in YANG Designer 56
Figure 45: Selecting the YANG module to open into YANG Designer..... 56
Figure 46: YANG Designer prompts you to provide the location of the required module..... 57
Figure 47: Specifying the location of the imported module in the Open Module dialog box..... 58
Figure 48: Opened module is displayed in YANG Designer for further editing..... 59
Figure 49: The Known Modules dialog box (viewing details of selected module)..... 60
Figure 50: Specifying a different include file in the Known Modules dialog box 61
Figure 51: User overridden include submodule is displayed in blue..... 61
Figure 52: Loading selected modules from the Known Modules dialog box..... 61
Figure 53: The Module from the Known Modules dialog box opens in YANG Designer 62
Figure 54: Validation messages displayed in the Errors tab of the Output window..... 64
Figure 55: Double-click a message to quickly locate the corresponding node in the YANG Tree (Design mode) 64
Figure 56:Viewing a module's resolved YANG tree with expanded YANG statements..... 65
Figure 57: Exporting a finished module to YANG file format..... 66
Figure 58: Exporting a submodule to a YANG format..... 67

1 INTRODUCTION

This guide contains instructions for completing basic operations in MG-SOFT Visual YANG Designer Professional Edition application. Majority of instructions are provided on a step-by-step basis, which should help the reader start using the software effectively. This guide is not meant as the ultimate reference guide to designing YANG modules that would describe the meaning and usage of every existing YANG statement. Instead, it describes the basic methods and principles of how YANG modules and submodules are designed in Visual YANG Designer. For complete reference about the YANG data modeling language, please refer to the YANG specification (RFC 6020).

Visual YANG Designer features two modes of operation: **Design mode** and **Source mode**. Majority of instructions in this document refer to the Design mode, which is the basic of operation. General guidelines for designing YANG modules in the advanced Source mode of operations are provided in section Design New YANG Module in Source Mode.

It is supposed that you are familiar with using a graphical computer environment, such as choosing a main menu command or a pop-up command, selecting items, closing windows and dialog boxes, using the drag&drop technique, etc.

All program commands in this manual are written in bold and italic letters. Individual commands in combinations of commands are separated by the "/" character. For example: ***Edit / Preferences*** – which means: click the "Edit" entry in the menu bar and select the "Preferences" command from the "View" menu.

All hyperlinks in text are marked with blue colored letters, e.g., Starting Visual YANG Designer. Clicking a hyperlink opens the page, which the hyperlink points to.

1.1 Product Description

MG-SOFT Visual YANG Designer lets you design, edit and validate NETCONF YANG and YIN modules in a visual manner, without having to master the YANG or YIN syntax. In addition to the **visual design mode** of operation, the software now features also an advanced **source mode** of operation, which incorporates a full-blown YANG source code editor that includes syntax coloring and intelligent code completion feature. The source mode of operation boosts productivity for experienced YANG authors who prefer writing the YANG code themselves with convenience of the built-in YANG authoring mechanisms, like autocomplete, find usages, go to definitions, find/replace, etc.

The software can load existing YANG and YIN modules and let you edit them either in a visual manner (where module elements are graphically represented as nodes in a hierarchical tree view) or by using the built-in YANG source code editor.

To design a YANG/YIN module in a visual mode of operation, simply drag&drop YANG nodes (e.g., container, list, leaf, leaf-list, etc.) from the components toolbar to the desired position in the module tree and set their properties in the Node Properties panel of the main window.

To import other YANG/YIN modules and reference their definitions in the current module, use the "File / Import" command. This will import the selected YANG/YIN module and display its tree in the Components window panel - Imports tab.

The statements in the imported module may then be referenced in the current module, e.g., nodes in the current module can use the typedefs defined in the imported module. To use other nodes from the Imports tab, select the node you want to use (e.g., grouping, extension, feature, etc.) and drag&drop it to the proper position in the current module displayed in the YANG Tree window panel in the left section of the main window.

The software includes YANG consistency validator and lets you export finished modules to the NETCONF native YANG format or to XML-based YIN syntax format.

MG-SOFT Visual YANG Designer is a Java™ application that can be installed and used on Windows, Linux and Mac OS X operating systems with Java Runtime Environment version 6.0 (a.k.a. JRE 1.6) or later installed.

2 INSTALL VISUAL YANG DESIGNER PROFESSIONAL EDITION

This section presents the basic system requirements your computer has to meet to install and use MG-SOFT Visual YANG Designer Professional Edition, and it describes the installing procedures for MG-SOFT YANG Designer on Windows, Linux, Mac OS X operating systems.

2.1 Requirements

MG-SOFT YANG Designer is a Java(TM) application that can be installed and used on Windows, Linux and Mac OS X operating systems with **Java Runtime Environment version 6.0 (a.k.a. JRE 1.6) or later** installed.

Additionally, administrator/root user privileges are required to install the software.

2.1.1 Windows Operating System

The Windows version of MG-SOFT YANG Designer has been tested on the following 32-bit and 64-bit Microsoft Windows operating systems:

- Windows XP,
- Windows Server 2003,
- Windows Vista,
- Windows Server 2008,
- Windows 7,
- Windows Server 2012,
- Windows 8.x.

2.1.2 Linux Operating System

The Linux version of MG-SOFT YANG Designer has been tested on the following 32-bit and 64-bit Linux distributions running on Intel x86/x86_64 architecture:

- Red Hat Enterprise Linux 5 or newer,
- Fedora Core 8 or newer,
- SUSE 11 or newer,
- Ubuntu 9 or newer,
- Slackware 13 or newer.

For the most recent information about the supported distributions, please refer to the release notes (READ_ME.TXT) of the current software release.

2.1.3 Mac OS X Operating System

The Mac OS X version of MG-SOFT YANG Designer has been successfully tested by MG-SOFT on:

- Mac OS X v10.6.x Snow Leopard (Intel)
- Mac OS X v10.7.x Lion (Intel)
- Mac OS X v10.8.x Mountain Lion (Intel)
- Mac OS X v10.9.x Mavericks (Intel)

2.2 Installing Visual YANG Designer

Before you install Visual YANG Designer Professional Edition on your computer, first make sure your computer meets the system requirements described in the Requirements section.

2.2.1 Windows Operating System

1. Put the MG-SOFT YANG Designer Professional Edition CD into your CD or DVD drive.

Note: To install the software on Windows, you need to have administrative privileges.

2. Click the ***Start*** button and select the ***Run*** command.
3. The Run dialog box appears (Figure 1).

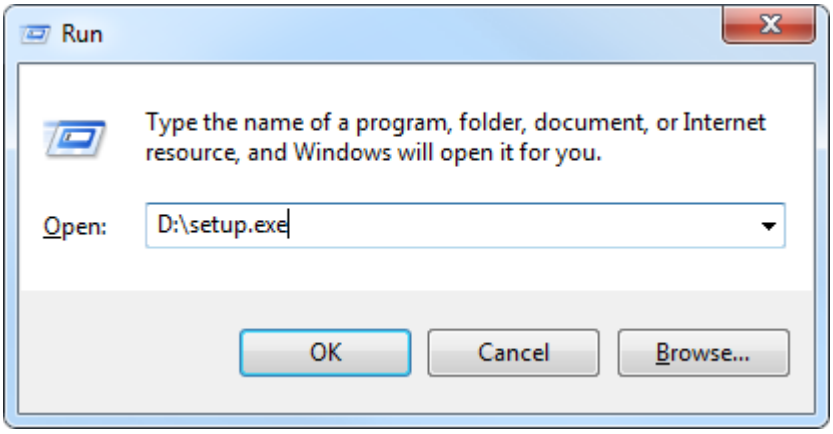


Figure 1: Run dialog box

4. Into the ***Open*** input line, type `D:\setup` and click the ***OK*** button.

Note: `D` is the letter assigned to the CD or DVD drive. If your CD or DVD has a different letter assigned, type that one instead of `D`.

5. Follow the instructions displayed on the screen.

Once the installation is complete, you can start MG-SOFT YANG Designer program.

2.2.2 Linux Operating System

The root user privileges are required to install the software.

Before the installation, please close all running MG-SOFT applications and uninstall any previous version of MG-SOFT Visual YANG Designer Professional Edition from the system.

1. Put the MG-SOFT Visual YANG Designer Professional Edition CD into your CD-ROM drive and mount the CD.
2. The software comes in three different software packages (`rpm`, `deb` and `tgz`). Depending on your Linux distribution, run one of the following commands in a Terminal window to install the software:

a) Linux distributions with the **RPM** package manager (RHEL, Fedora, SUSE, Mandriva, etc):

- On a 32-bit Linux distribution with the RPM package manager, install the 32-bit (i386) RPM package:

```
# rpm -ivh mgYangDesigner_2015-X.X-X.i386.rpm
```

- On a 64-bit (x86_64) Linux distribution with the RPM package manager, install the RPM package containing the 64-bit build of the software, as follows:

```
# rpm -ivh mgYangDesigner_2015-X.X-X.x86_64.rpm
```

b) Linux distributions with the **DPKG** package manager (Debian, Ubuntu, etc.):

- On a 32-bit Linux distribution with the DPKG package manager, install the 32-bit (i386) DEB package:

```
# dpkg -i mgYangDesigner-2015_X.X-X_i386.deb
```

- On a 64-bit (x86_64/amd64) Linux distribution with the DPKG package manager, install the corresponding DEB package, as follows:

```
# dpkg -i mgYangDesigner-2015_X.X-X_x86_64.deb
```

c) Linux distributions with the **installpkg** package manager (e.g., Slackware):

- On a 32-bit (i386) Linux distribution with the installpkg package manager, install the 32-bit TGZ package:

```
# installpkg mgYangDesigner_2015-X.X-i386-X.tgz
```

- On a 64-bit (x86_64) Linux distribution with the installpkg package manager, install the corresponding TGZ package:

```
# installpkg mgYangDesigner_2015-X.X-x86_64-X.tgz
```

In case you have KDE or GNOME Environments installed on your machine, the installation will add an entry to the K Menu or Gnome Menu respectively.

Once the installation is complete, you can start MG-SOFT YANG Designer program.

2.2.3 Mac OS X Operating System

1. Double-click the MG-SOFT YANG Designer disk image file (`.dmg`) that you have downloaded from MG-SOFT's Website or obtained on a removable medium.

   Tip: Use **Finder** to navigate to the DMG file if it is not located on your desktop.

2. The contents of the double-clicked disk image displays in a Finder window. MG-SOFT Visual YANG Designer virtual drive appears on the desktop. Drag&drop the "MG-SOFT YANG Designer.app" to the "Applications" folder within the MG-SOFT Visual YANG Designer virtual drive.

Once the installation is complete, you can start MG-SOFT YANG Designer for Mac OS X from the Finder.

3 START VISUAL YANG DESIGNER PROFESSIONAL EDITION

3.1 Starting Visual YANG Designer

3.1.1 Windows Operating System

1. In Windows operating systems, select the ***Start*** / ***Programs*** / ***MG-SOFT YANG Designer*** / ***YANG Designer*** command from the Windows taskbar.
2. The YANG Designer desktop appears and you can start using the software. Please refer to the Apply License Key section for instructions on how to apply your license.

3.1.2 Linux Operating System

The easiest way to start YANG Designer under Linux operating system is to use the start menu. The start menu can be displayed from the desktop taskbar.

1. If you have the KDE or GNOME desktop environment installed, display the ***K/Gnome*** start menu by clicking the button in the left corner of your taskbar.
2. To start YANG Designer, search for and use the ***MG-SOFT YANG Designer*** / ***YANG Designer*** command.
3. The YANG Designer desktop appears and you can start using the software. Please refer to the Apply License Key section for instructions on how to apply your license.

> **Tip:** To start the software from the command prompt, open the Terminal window, CD to the directory where the "mgYangDesigner.jar" file is (/usr/local/mg-soft/mgyangdesigner/java/) and run the following command:
>
> ```
> # java -jar mgYangDesigner.jar
> ```

3.1.3 Mac OS X Operating System

1. Open the **Finder** and select the ***Applications*** entry in the panel on the left.
2. Select and double-click the "MG-SOFT YANG Designer.app" icon to start the YANG Designer application.
3. The YANG Designer desktop appears and you can start using the software. Please refer to the Apply License Key section for instructions on how to apply your license.

3.2 Visual YANG Designer Desktop

The Visual YANG Designer desktop is composed of typical graphical user interface components, like the title bar, menu bar, toolbar and the working area.

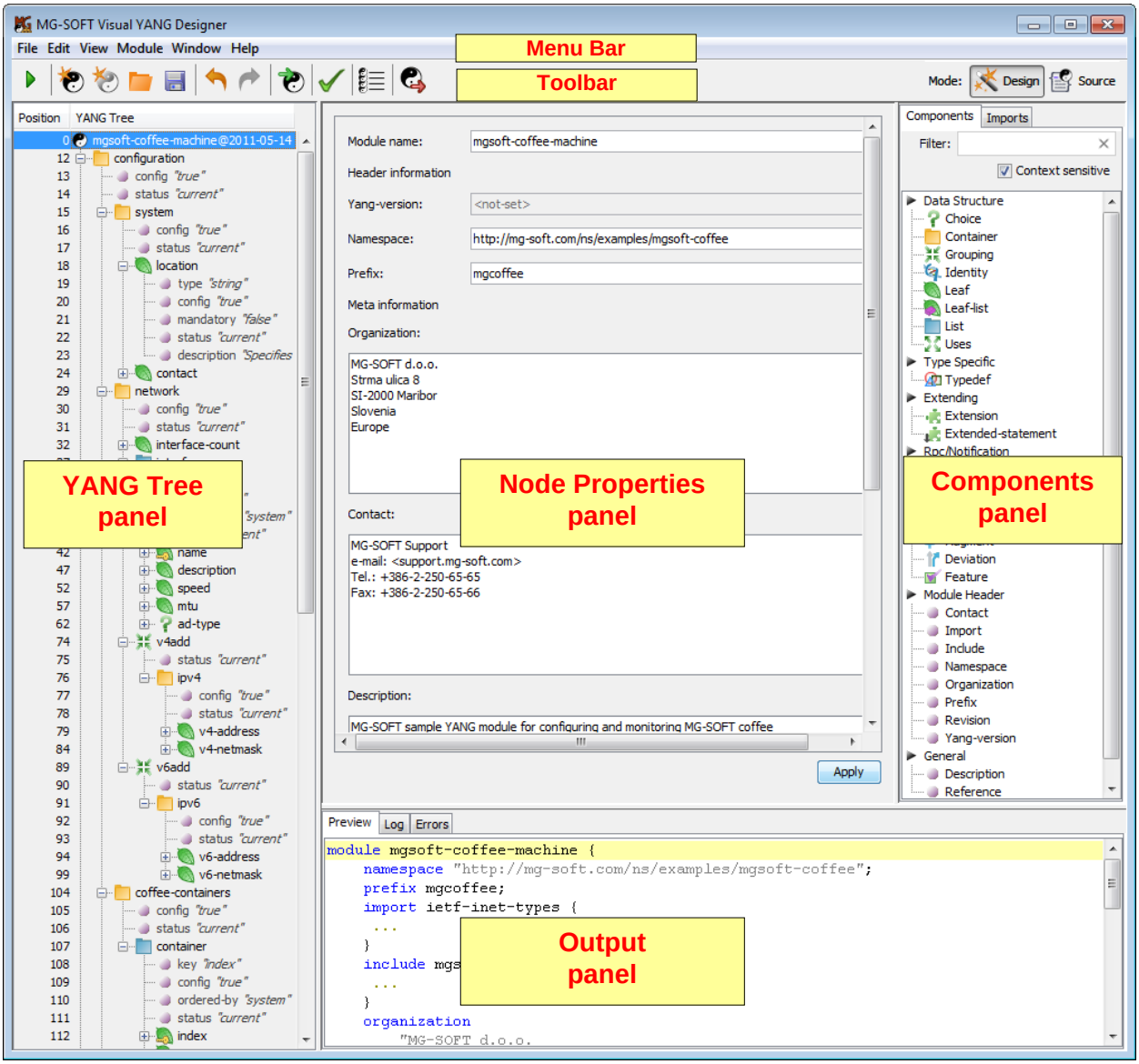


Figure 2: Visual YANG Designer desktop - Design mode

3.2.1 Two Modes of Operation: Design Mode and Source Mode

Visual YANG Designer features two modes of operation: **Design mode** and **Source mode**. Both modes let you achieve the same goal: create, edit and validate YANG modules and submodules, but they let you achieve that goal in a different way. The Design mode is a visual mode, where you design a graphically presented YANG tree model by using the appropriate GUI controls and letting the software generate the resulting YANG code for you. The Source mode of operation, on the other hand, is an advanced, text-based mode, where you design YANG modules and submodules by writing the YANG code directly in the built-in YANG source code editor.

The default mode of operation is Design mode. However, you can switch between the Source and Design mode at any time and continue designing your YANG module in the currently selected mode.

> **Note:** YANG source code is re-generated in the background after you make (and apply) any change to the module in Design mode. Visual YANG Designer always generates YANG code with statements in **canonical order** as per RFC 6020, section 12.
>
> If you had previously created or edited a YANG module in Source mode or in a third-party application and you then made some modifications to it in Design mode, this may change the order of statements in your YANG code (i.e., the canonical order of statements will be applied in the code).

To switch between Design and Source mode, select the desired mode from the ***View*** menu or click the corresponding mode selection button in the right section of the toolbar (Figure 3).

Figure 3: Selecting the desired mode of operation in Visual YANG Designer

Depending on the mode of operation selected, different window panels are displayed in the Visual YANG Designer working area, as follows:

Design Mode

Design mode is the default mode of operation. It lets you design NETCONF YANG modules and submodules in a visual manner by placing graphically presented YANG statements (nodes) into the hierarchically structured YANG tree and setting their properties in the accompanying window panel. In the background, the software automatically generates the corresponding YANG code based on the visual edits you make.

To activate the Design mode, select the ***View / Design Mode*** from the menu or click the ***Design mode*** toolbar button (

).

The following window panels form the **Design mode** working area (Figure 2):

- **YANG Tree** (left panel)
  The YANG Tree panel contains two columns:

  Position (column)
  Displays numbers that indicate the position of each node in the YANG tree relative to the module root node or the corresponding submodule root node. Positions are referenced by the error and warning messages displayed in the Errors tab of the Output panel (Design mode).

  YANG Tree (column)
  Displays the hierarchical tree structure of the YANG module and submodules you are viewing/designing, where module elements (statements/substatements) are graphically represented as nodes in a hierarchical tree. You can reposition the nodes in the tree using the drag&drop technique.

- **Node Properties** (middle panel)
  Displays and lets you configure the properties of the node selected in the YANG Tree panel.

- **Components** (right panel)
  Displays the components that are used as building blocks for creating a module tree. The components are YANG statements/substatements (**Components tab**) and nodes from imported modules (**Imports tab**). By default, the YANG statements/substatements are context sensitive, meaning that only those YANG statements/substatements are displayed that can be used with the currently selected node in the YANG Tree panel. To design a YANG module, simply drag components from the Components panel, drop them to the desired position in the YANG Tree panel and set the node properties in the Node Properties window panel.
- **Output** (bottom panel)
  The Output panel contains three tabs:

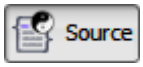
  Preview tab
  Displays the definition (in YANG language) of the node currently selected in the YANG Tree panel.

  Log tab
  Displays application log records that indicate module compilation, loading and validation progress.

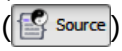
  Errors tab
  Displays module validation messages (Error/Warning).

The above listed window panels are arranged side-by-side in the main window. Window panels can be resized by dragging their borders. Additional windows and dialog boxes can be opened from the program menu, toolbars and the pop-up (context) menus.

Source Mode



Source mode is an advanced mode of operation, where you design NETCONF YANG modules by writing the YANG code directly in the full-fledged YANG source code editor, which incorporates syntax coloring and intelligent code completion. The Source mode is suitable for more experienced YANG authors who prefer writing YANG code themselves.

To activate the Source mode, select the ***View / Source Mode*** from the menu or click the ***Source mode*** toolbar button (

).

The following window panels form the **Source mode** working area (Figure 4):

- **YANG Tree Preview** (upper-left panel)
  Displays the hierarchical tree structure of the YANG module and submodules you are writing/designing, where module elements (statements/substatements) are graphically represented as nodes in a hierarchical tree, similar to the YANG Tree panel in the Design mode. The YANG tree preview is automatically built from the YANG code as you type it in the **YANG Editor** panel.

  Double-click a node in the YANG Tree Preview panel to display its definition in the YANG Source Editor panel.
- **Imports** (lower-left panel)
  Displays the imported modules and their hierarchical structure.

- **YANG Source Editor** (upper right panel)

  Line numbers (gutter)
  Displays the code line numbers. Line numbers are referenced by the error and warning messages displayed in the Errors tab of the Output panel (Source mode).

  The gutter displays also error (🔴) and warning (⚠) symbols in front of the lines that contain validation inconsistencies (errors, warnings). Hover the mouse pointer over the error or warning symbol to view the respective validation message in a tooltip.

  YANG Editor (text editor)
  Displays and lets you edit the source code of the YANG module and submodules you are designing. YANG Editor incorporates the YANG syntax coloring and YANG keyword and argument auto-complete feature (activated by pressing the **CTRL+SPACE** keys).

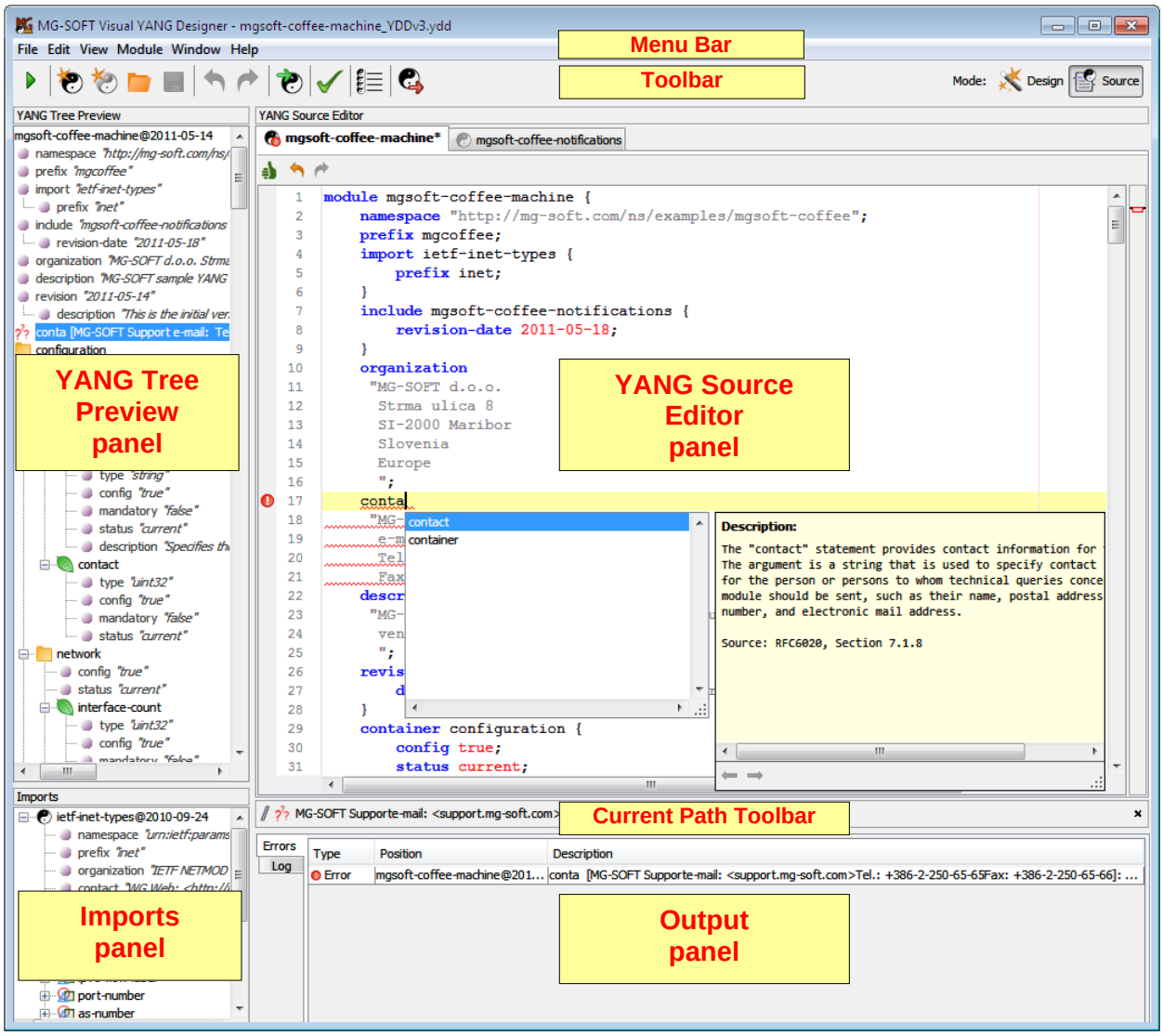


Figure 4: Visual YANG Designer desktop - Source mode

Page margin is displayed as a thin red vertical line in the right section of the YANG Editor. Page margin is set by default at 80. character when using a fixed-width font and can be configured in the program preferences (Source Editor view).

Annotation stripe (vertical ruler)
Annotation stripe is the vertical bar displayed next to the vertical scrollbar along the right border of the YANG Editor. The current line marker ( ) in the annotation stripe indicates the relative position of the cursor (its line number) in the document. The annotation stripe also displays relative locations (by line numbers) of detected errors ( ) and warnings ( ) and the locations of the found text ( ) when using the Find feature. Hover your mouse pointer over any mark in the annotation stripe to view more information in a tooltip (line number or error/warning message). Click a mark in the annotation stripe to quickly jump to the line in the document, referenced by the given mark.

- **Current Path Toolbar** (below YANG Source Editor panel)
  The Current Path toolbar displays the current YANG statement's absolute path in the module, as presented in the YANG Tree Preview panel (e.g., /configuration/network/interface/name). Click an element in the path (e.g., interface) to view its definition in the YANG Source Editor panel.

- **Output** (lower right panel)
  The Output panel contains two tabs by default:

  Errors tab
  Displays module validation messages (Error/Warning).

  Log tab
  Displays application log records, which indicate module compilation, loading and validation progress.

The above listed window panels are arranged side-by-side in the main window. Window panels can be resized by dragging their borders. Additional windows and dialog boxes can be opened from the program menu, toolbars and the pop-up (context) menus.

4 APPLY LICENSE KEY

To use MG-SOFT Visual YANG Designer without limitations, you need to apply a valid license.key file to the software, as follows:

1. Select the ***Help / Apply License Key*** command from the main menu or click the ***Apply License*** toolbar button (the latter is displayed only when the software is run without a valid license key file in place).
2. A dialog box appears that lest you select and apply your license key (Figure 5).

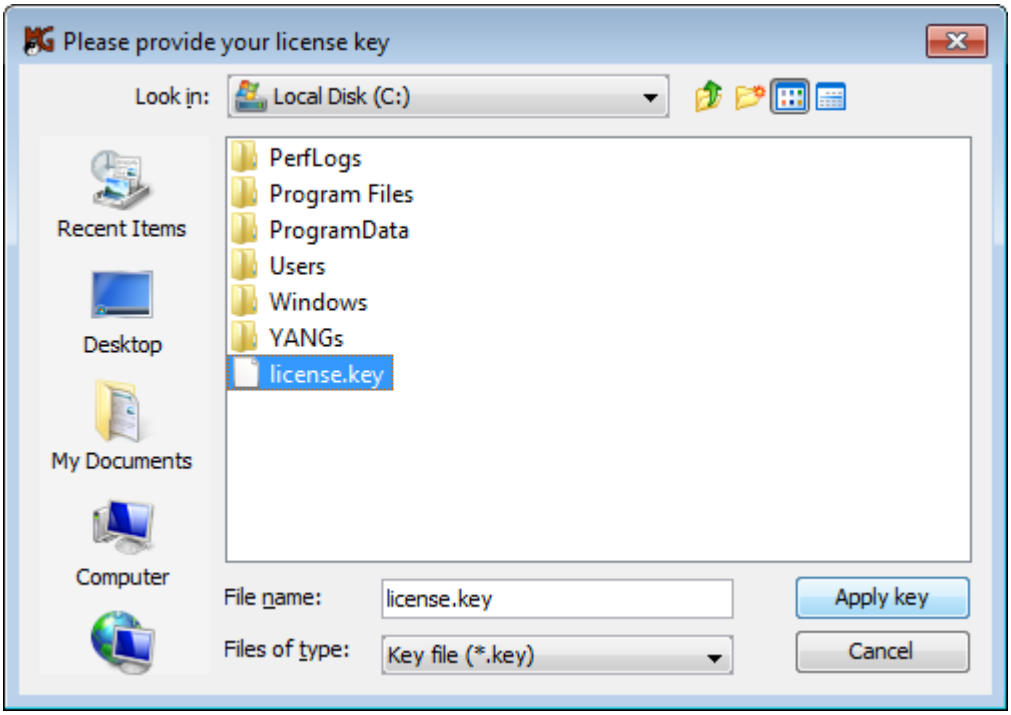


Figure 5: Selecting and applying the license.key file

3. Navigate to the drive and folder containing your `license.key` file for MG-SOFT Visual YANG Designer Professional Edition, select the `license.key` file and click the ***Apply License*** button in the license key selection dialog box.
4. The software will copy the selected `license.key` file to the proper location in order for Visual YANG Designer to read it and unlock its features respectively after a restart.
5. Exit Visual YANG Designer by choosing the ***File / Exit*** command and restart it, as described in the Starting Visual YANG Designer section. Now, the selected license should be applied and you can start using the software without licensing restrictions.

> **Tip:** You can check if the license key has been properly applied by verifying if the About Visual YANG Designer dialog box (accessible via the ***Help / About*** command) displays your license details correctly.

5 DESIGN NEW YANG MODULE IN DESIGN MODE

MG-SOFT Visual YANG Designer features two modes of operation: **Design mode** and **Source mode**.

Instructions in this section refer to the **Design mode**, which is the default mode of operation. Basic guidelines for designing YANG modules in the Source mode are provided in section Design New YANG Module in Source Mode.

If Design mode is not active, select the ***View / Design Mode*** from the menu or click the ***Design mode*** toolbar button (

) to activate it.

The Design mode lets you design and edit NETCONF YANG modules and submodules in a visual manner, where module elements are graphically represented as nodes in a hierarchical tree structure.

> **YANG** is a data modeling language for the Network Configuration Protocol (NETCONF). YANG is specified in RFC 6020. This document quotes and summarizes RFC 6020 on several places, including in the paragraphs below.
>
> **YANG module** defines a hierarchy of data that can be used for NETCONF-based operations, including configuration, state data, Remote Procedure Calls, and notifications. Typically, a YANG module defines a tree of data elements that represent the configuration and runtime status of a particular network element managed via NETCONF. A YANG module is usually stored in a file with the .yang file name extension.
>
> YANG modules can be translated into an equivalent XML syntax called **YIN** (**Y**ANG **I**ndependent **N**otation), allowing applications using XML parsers to operate on the models. The conversion from YANG to YIN is lossless. Typically, a YIN module is stored in a file with the .yin file name extension.
>
> **Submodules** are partial modules that contribute definitions to a module. A module may include zero or more submodules, but each submodule may belong to only one module.

To design a new YANG/YIN module in the visual mode of operation, first create a new module node, then drag&drop additional YANG nodes from the Components toolbar to the tree and set their properties in the MG-SOFT Visual YANG Designer GUI, as described in this section.

In the background, the software automatically generates the corresponding YANG code based on the visual edits you make.

5.1 Creating a New Module

1. In the main window, choose the ***File / New / Module*** command.
2. The New Module dialog box appears (Figure 6), which prompts you to enter the basic properties for the new YANG module.

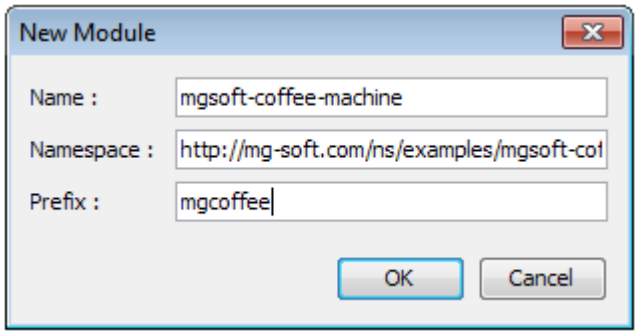


Figure 6: Specifying the basic parameters for a new YANG module

3. Into the ***Name*** input line, enter a name for the new YANG module you are creating (e.g., `mycompany-product1`).

> Private module names are assigned by the organization owning the module. It is recommended to choose module names that will have a low probability of colliding with standard or other enterprise modules and submodules, e.g., by using the enterprise or organization name as a prefix for the module name.

> **Identifiers** are used to identify different kinds of YANG items (including module nodes) by name. Each identifier starts with an uppercase or lowercase ASCII letter or an underscore character, followed by zero or more ASCII letters, digits, underscore characters, hyphens, and dots.
>
> An identifier must not start with ('X'|'x') ('M'|'m') ('L'|'l').
>
> Identifiers may contain up to 64 characters.
>
> Identifiers are **case sensitive**.

4. Into the ***Namespace*** input line, enter a URI to be used as the namespace for the new module (e.g., `http://mycompany.examples.com/p1.html`).

> All YANG definitions are specified within a module that is bound to a particular XML namespace, which is a globally unique URI (Uniform Resource Identifier). A NETCONF client and server use the namespace during XML encoding of data.
>
> Namespaces for private modules are assigned by the organization owning the module without a central registry. Namespace URIs must be chosen so they cannot collide with standard or other enterprise namespaces, for example, by using the enterprise or organization name in the namespace.

5. Into the ***Prefix*** input line, enter a prefix string for the new module (e.g., `mycompany-p1`).

> The "prefix" statement is used to define the prefix associated with the module and its namespace. The "prefix" statement's argument is a string that is used as a prefix to access a definition contained in the module, e.g., "`if:ifName`", where `if` is the prefix of the module and `ifName` is a node defined within that module.
>
> When used inside the "module" statement, the "prefix" statement defines the prefix to be used when this module is imported, unless there is a conflict.

6. After you have specified the above settings, click the ***OK*** button to create the new module and close the New Module dialog box. A new node of type **module** ( ) is added to the empty YANG Tree window panel displayed on the left-hand side of the of the MG-SOFT YANG Designer's main window and those properties of the module node that are set to some value (namespace, prefix, revision) are displayed as its sub-nodes ( ). Note that for convenience, the **revision** property of the module node is set automatically to the current

system date (you can edit it to match your preferences). The newly added module node is automatically selected and its main properties are displayed also in the Node Properties window panel in the center of the main window (Figure 7).

The Node Properties window panel displays the main properties of the node currently selected in the YANG Tree panel. The main properties are those which have **cardinality of 1 and 0..1**. The only exception to this rule is the ***revision*** property of the ***module*** node, which has cardinality of 0..n and is also displayed Node Properties window panel. The cardinality of all YANG (sub)statements is specified in the YANG specification.

Cardinality designates how many instances of the given property type are valid for the given node. For example, cardinality of 1 means that the property is mandatory, i.e., exactly one instance of this property must be present for the given type of node (e.g., ***namespace*** property of the ***module*** node). The cardinality of 0..1 means that exactly zero or one instance of this property type may exist (e.g., ***description*** property of the ***leaf*** node). Properties of other cardinalities (e.g., 0..n) which are not displayed in the Node Properties window panel can be manually added to the node by dragging the corresponding property from the Components window and dropping it onto the node in the YANG Tree panel.

Once the value of a node property has been set, this property appears as a sub-node ( ) in the YANG tree panel (e.g., ***namespace***, ***prefix***, ***revision*** property for the ***module*** node).

Tip: If you want to disable displaying property sub-nodes for the entire YANG tree, check the **Hide property nodes** checkbox in the YANG Designer Preferences dialog box (**Edit/Preferences**), Design Mode view.

To hide or display the properties (sub)nodes for all nodes in the selected subtree, right-click the subtree node and choose the **View Property Nodes / Hide for Subtree** or the **View Property Nodes / Show for Subtree** pop-up command, respectively.

To hide or display the properties (sub)nodes of the selected node only, right-click the node and choose the **View Property Nodes / Toggle for Selected Node** pop-up command.

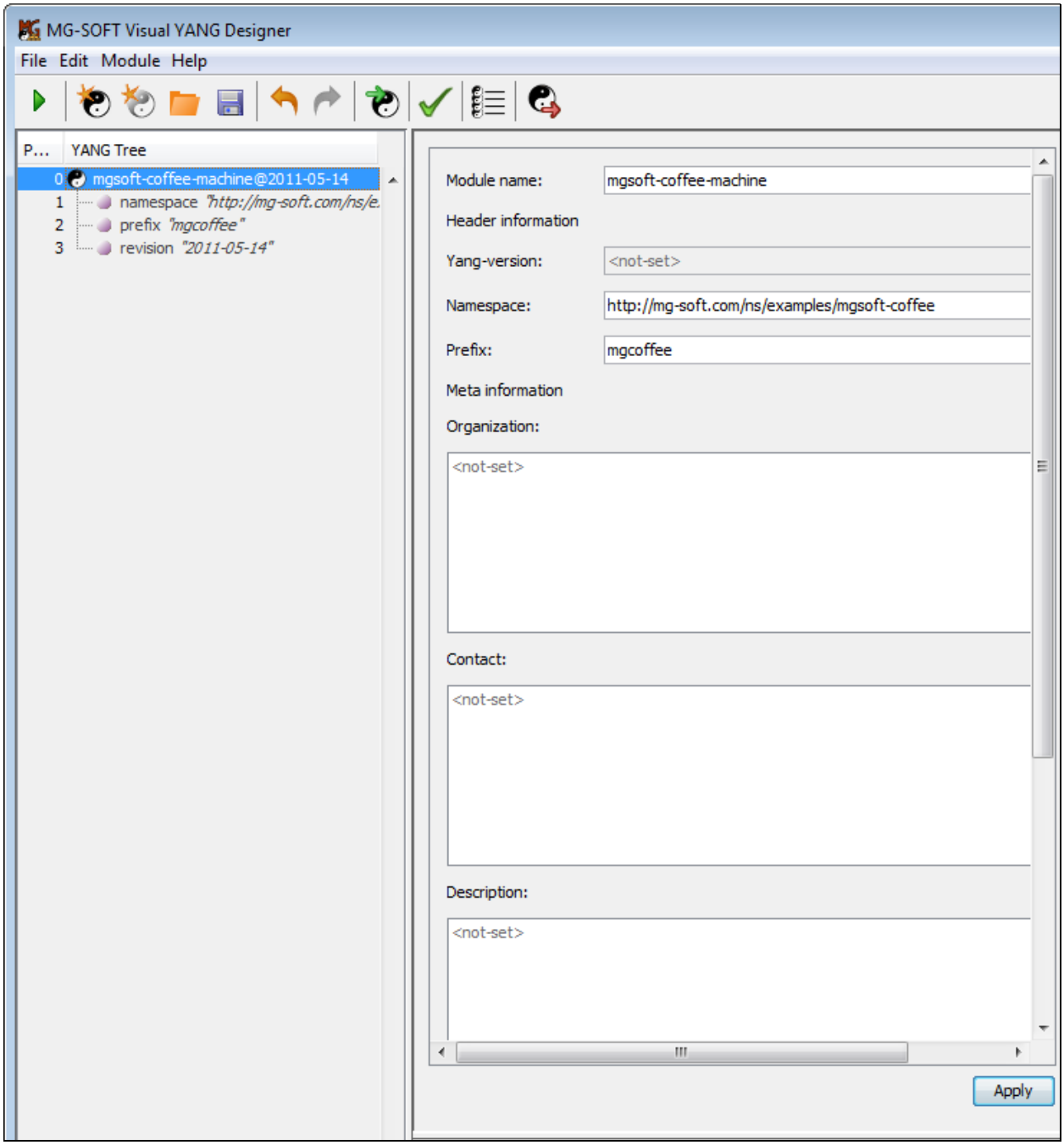


Figure 7: A newly added module node and its properties

7. The remaining (optional) properties of the **module** node (e.g., Organization, Contact, Description, Reference, etc.), which have no value set, are marked as **`<not-set>`**. The property sub-nodes ( ) of the properties that do not have any value set are not shown in the YANG tree. To configure the remaining properties of the module node in the Node Properties window panel, proceed as follows:

> For the exact meaning of the remaining **module** node properties (substatements), please refer to the YANG specification, section 7.1 (http://tools.ietf.org/html/rfc6020#section-7.1).

1. Click into the corresponding input field to enable editing it (the `<not-set>` mark disappears).

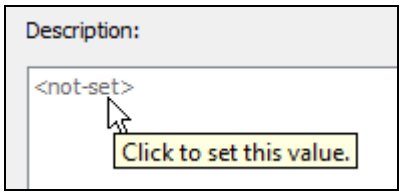


Figure 8: Click a disabled input field of a module property to enable setting its value

2. Enter the text into the input field to set its value. To set the value to empty (zero-length) string, simply leave the input field blank (when applicable).

> **Tip:** You can quickly set any property value back to the **<not-set>** state by holding down the **CTRL** button and **clicking** the corresponding property control (e.g., input line, drop-down list, etc.)

3. Click the ***Apply*** button at the bottom of the Node Properties window panel to apply the changes (Figure 9). After clicking the ***Apply*** button, the newly set property appears as a sub-node ( ) in the YANG Properties panel.

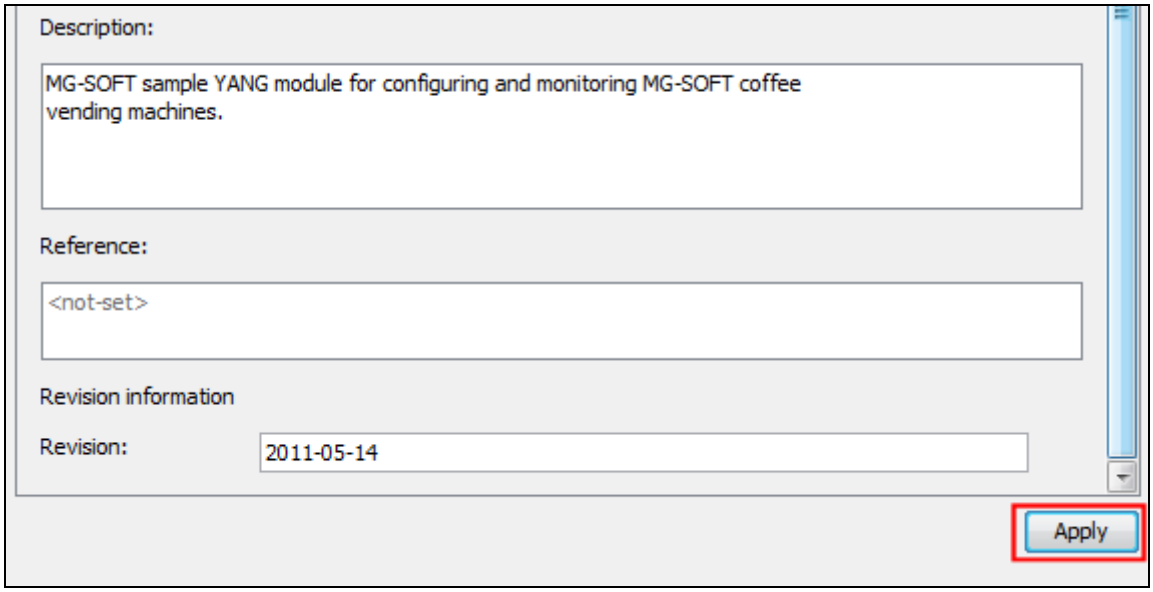


Figure 9: Applying changes made in the Node Properties window panel

8. To add a property that is not displayed in the Node Properties window panel for the module node or to add an additional instance of a property (property with cardinality of 0..n), drag the desired property from the Components window and drop it onto the module node. For example, an additional **revision** property node can be added to the module node when the module is revised (Figure 10).

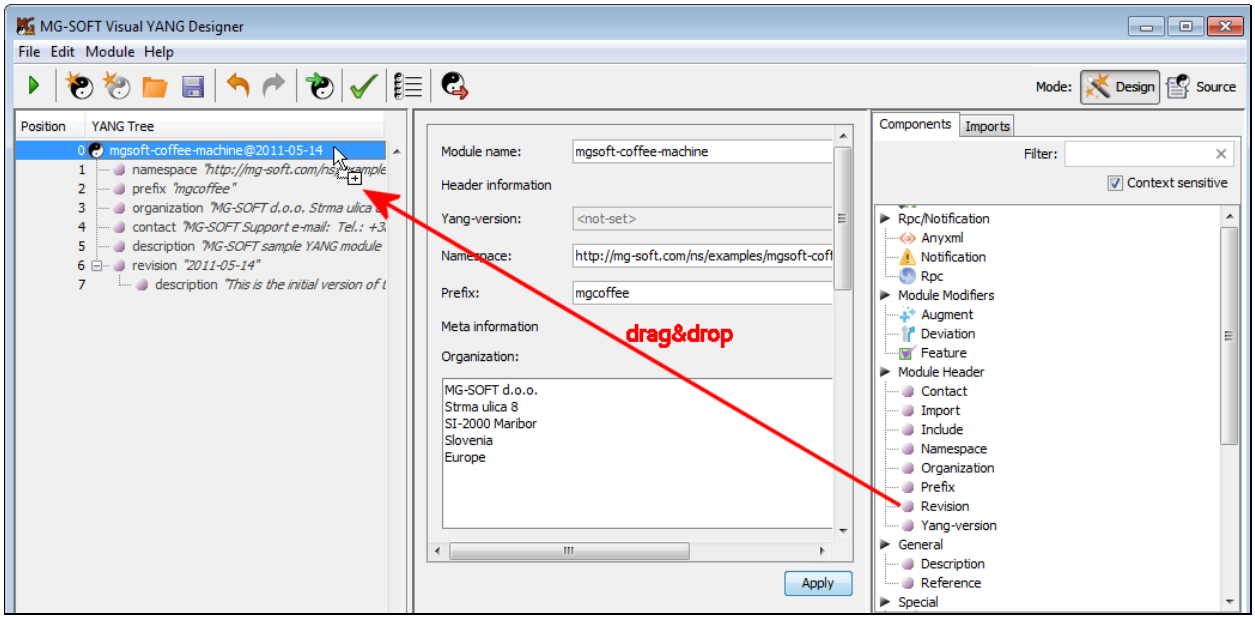


Figure 10: Adding an additional revision property to a module node

9. This adds a new property sub-node to the module node in the YANG Tree panel and displays its properties in the Node Properties panel. Select the property sub-node and configure its properties in the Node Properties panel and click the ***Apply*** button to apply the changes.

5.1.1 Adding Nodes to a New Module

YANG defines the following four types of nodes for data modeling:

Node icon in YANG Designer	Node type	Description
	leaf	A leaf node contains simple data like an integer or a string. It has exactly one value of a particular type and no child nodes.
	leaf-list	A leaf-list is a sequence of leaf nodes with exactly one value of a particular type per leaf.
	container	A container node is used to group related nodes in a subtree. A container has only child nodes and no value. A container may contain any number of child nodes of any type (including leafs, lists, containers, and leaf-lists).
	list	A list defines a sequence of list entries. Each entry is like a structure or a record instance, and is uniquely identified by the values of its key leafs. A list can define multiple key leafs and may contain any number of child nodes of any type (including leafs, lists, containers etc.)

To model the configuration and state data of a NETCONF device, you can take the above (and other) types of nodes that are available in the Components window panel, add them to the YANG tree and set their properties, as described in this section.

About configuration and state data nodes

> RFC 6241, section 1.4: The information that can be retrieved from a running system is separated into two classes, **configuration data** and **state data**. Configuration data is the set of writable data that is required to transform a system from its initial default state into its current state. State data is the additional data on a system that is not configuration data such as read-only status information and collected statistics.

In contrast to **configuration data** nodes, which are depicted with normal-colored icons, the **state data** nodes are depicted with light-colored (translucent) icons in the YANG Tree panel. This principle lets you quickly distinguish between configuration (e.g., read-write) and state data (read-only) nodes in the YANG tree. For example:

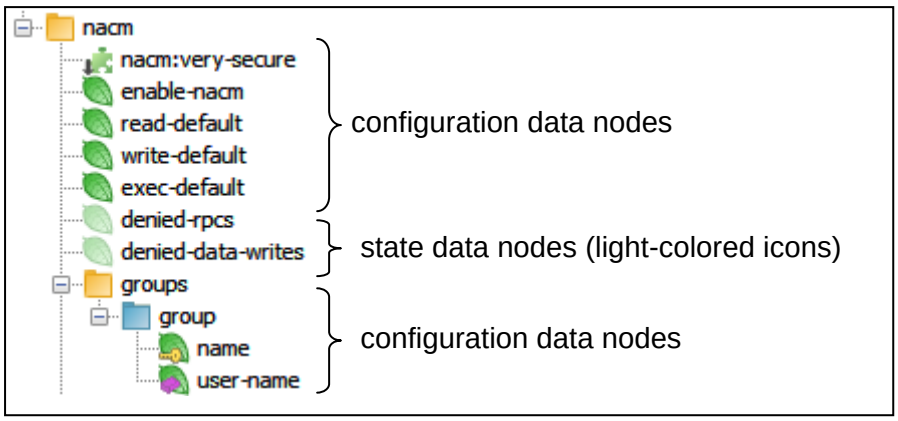


Figure 11: Example of configuration and state data nodes displayed in the YANG Tree panel

By default, nodes that you add to the YANG Tree are configuration data nodes (unless the parent node is a state data node). To set a node as a state data node, click its ***Config*** checkbox to enable it (by default it is in disabled state) and then leave it unchecked. This will set the node **Config** property to **`false`** (Figure 12). All child nodes (if any) of a state data node will also be state data nodes.

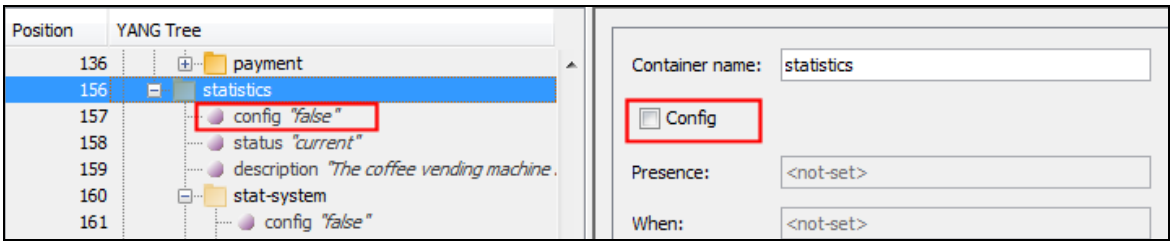


Figure 12: Example of a state data node with *Config* property set to “false” (empty *Config* checkbox)

Filtering Components in the Components Panel (Components Tab)

By default, the **Components** tab in the Components panel has context sensitive filter applied, meaning that it displays only those components (YANG statements) that can be added as child items to the currently selected node according to the YANG specification. Therefore, if the **module** ( ) node is selected in the YANG tree, The Components tab

displays only those types of components (YANG statements) that can be added as substatements to the module statement (Figure 13).

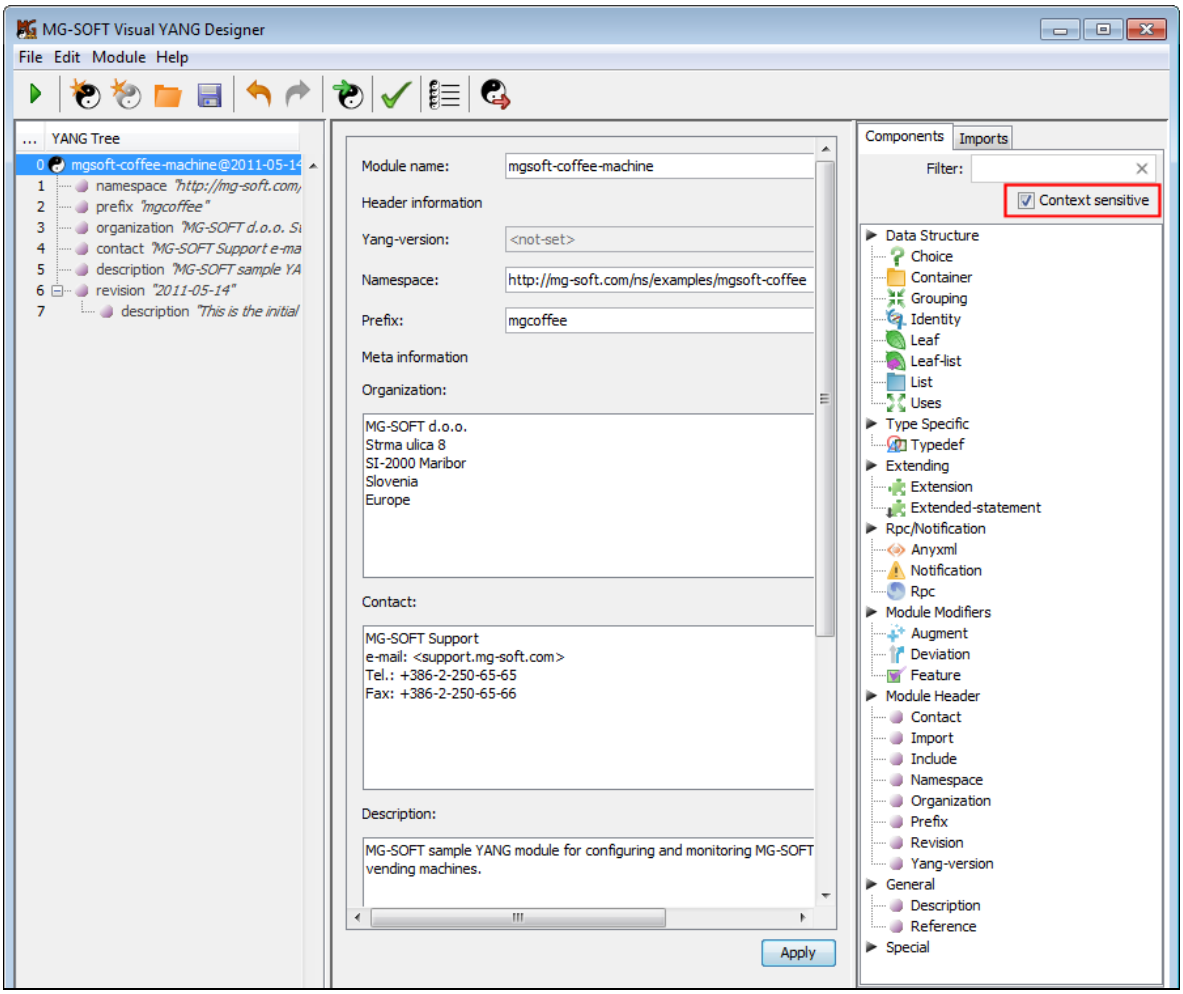


Figure 13: The Components tab in the Components panel is context sensitive by default

The context sensitive filter is convenient when building the YANG tree structure, as it enables easier finding of the valid components (YANG statement), based on the currently selected component.

If you want to disable the context sensitive filter and view all available components instead, uncheck the ***Context sensitive*** checkbox under the ***Filter*** input line in the Component panel (Figure 13).

To filter the components by your search phrase, enter the desired text into the ***Filter*** input line in order to display only those components that **include** the entered text. If the ***Context sensitive*** checkbox is checked, the entered filter phrase will search only within the components that satisfy the context sensitive filter.

Building a YANG Schema Tree

There are two ways to build a YANG schema tree in the Design mode:

- By dragging and dropping the components (nodes) from the Components window panel to the YANG Tree window panel, or
- By using the mouse context (pop-up) menu in the YANG Tree window panel.

This section describes both options of designing a YANG tree.

Building YANG Tree by Using Drag&Drop Technique

1. Select the **module** node in the YANG Tree panel. Leave the ***Context sensitive*** checkbox checked in the Components panel to display only those nodes that can be added as child items to the module node.
2. To add a node to the YANG/YIN module, select the desired node type (e.g., **container**) in the Components window panel (Components tab) displayed on the right hand-side of the Visual YANG Designer's main window, and drag-and-drop it onto the **module** node in the YANG Tree panel displayed in the left portion of the main window (Figure 14).

> **Note:** Nodes in the YANG Tree window panel can be positioned by using the drag-and-drop technique. As you drag the selected node along the tree, Visual YANG Designer indicates where the node will be inserted when you release the mouse button (not all node types can be positioned to all locations in the tree). There are 3 possibilities:
>
> * The target node is selected. This means that the node you are about to drop will be added as the last child node of the selected node.
> * A line is drawn between two nodes. This means that the node you are about to drop will be inserted between these two nodes.
> * None of the above. The node cannot be placed to the current location.
>
> In addition, Visual YANG Designer displays also a tooltip with a short description of the respective drop action (Figure 14) as you drag selected node along the YANG tree.

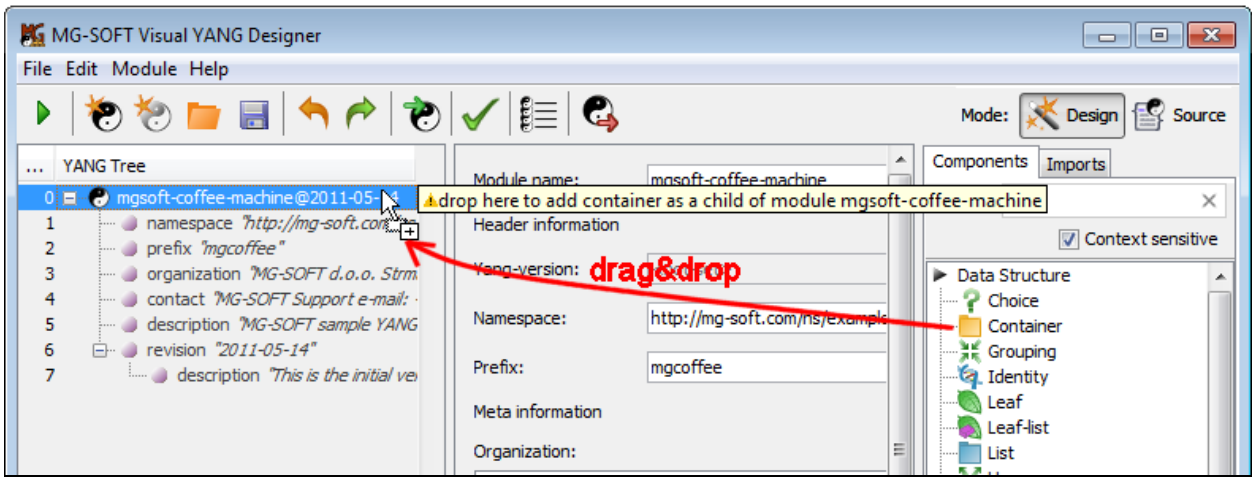


Figure 14: Adding a *container* node to *module* node using drag&drop technique

3. A new node of the selected type (e.g., **container**, **list**, **leaf**, **leaf-list**, etc.) is added as a child node of the **module** node in the YANG Tree window panel. The new node is

automatically selected and the Node Properties panel in the center of the main window displays its properties (default values are preset for some properties). The displayed properties depend on the type of the node selected in the tree (Figure 15).

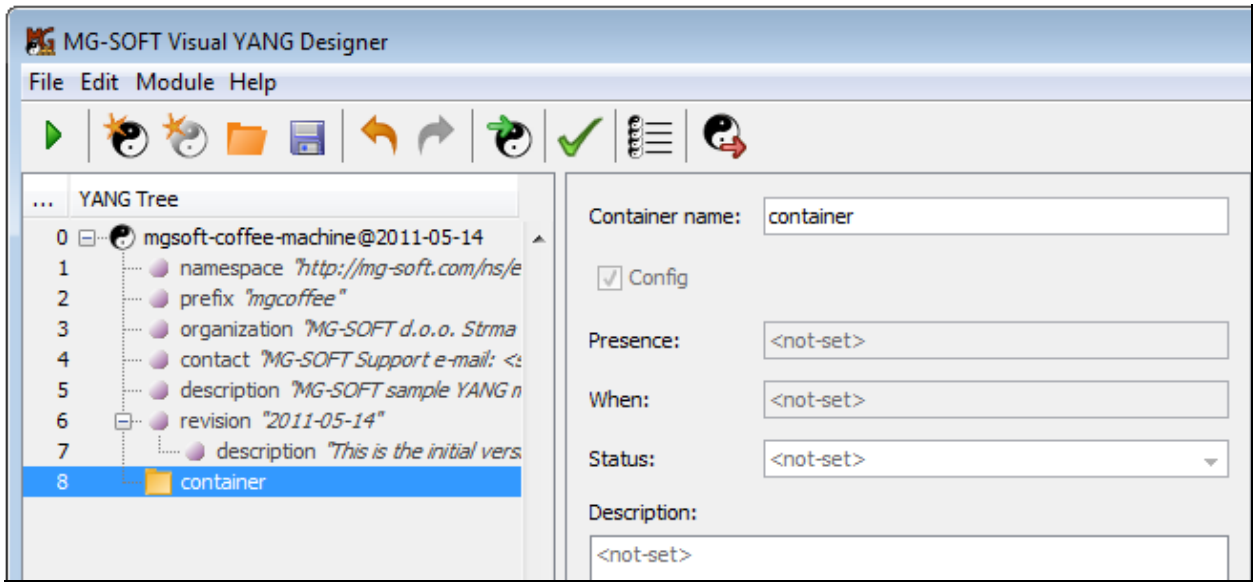


Figure 15: A newly added container node and (some of) its default properties

Building YANG Tree by Using "Add" Pop-up Menu Command

1. Select the **module** node in the YANG Tree panel.
2. Right-click the selected node and choose the desired node type to be added from the ***Add*** cascading pop-up menu. For example, to add a **container** node, select the ***Add / Data Structure / Container*** command from the mouse pop-up menu (Figure 16).

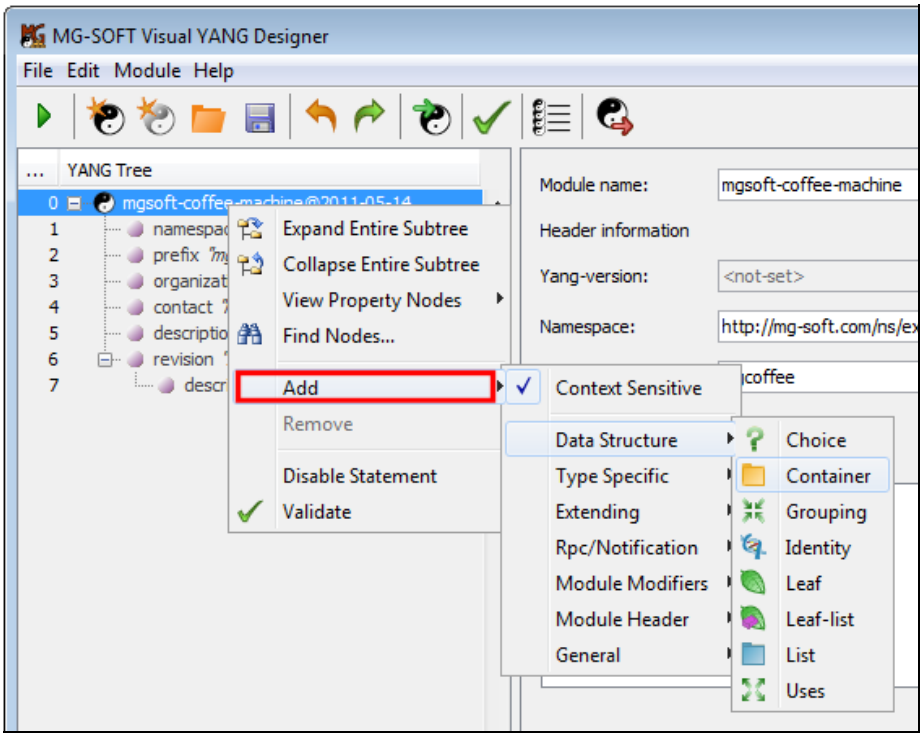


Figure 16: Adding a *container* node to *module* node using the pop-up menu

> **Tip:** Leave the ***Context sensitive*** toggle command selected in the ***Add*** cascading pop-up menu to display only those nodes that can be added as child items to the currently selected node.

3. A new node of the selected type (e.g., **container**, **list**, **leaf**, **leaf-list**, etc.) is added as a child node of the **module** node in the YANG Tree window panel. The new node is automatically selected and the Node Properties panel in the center of the main window displays its properties (default values are preset for some properties). The displayed properties depend on the type of the node selected in the tree (Figure 15).
4. Set the values of properties of the selected node in the Node Properties window panel (e.g., **Name**, **Config**, **Type**, **Status**, **Description**, etc.) and click the ***Apply*** button at the bottom of this panel to apply the changes (Figure 15). If a GUI component (drop-down list, input line, input field, checkbox,…) for setting a particular property is disabled or marked as `<not-set>`, click the given component to enable setting its value.

> **Tip:** If you make a mistake while editing the node properties or while rearranging the YANG tree, use the **Undo** feature (***Edit/Undo*** or the toolbar button) to quickly undo the edit and revert the item to its previous state. Use the **Redo** feature (***Edit/Redo*** or the toolbar button) to re-apply the undone edit.

5. To configure a property that is not displayed in the Node Properties window panel for the given node type or to add an additional instance of a property (property with cardinality of 0..n) to the given node, drag the desired property from the Components window panel and drop it onto the node (see an example in Figure 10).

> **Tip:** If the desired property component is not displayed in the Components window panel (Components tab), select the node which you want to add the property component to in the YANG tree panel or uncheck the ***Context sensitive*** checkbox under the ***Filter*** input line in the Component panel (Figure 13).

6. The corresponding YANG definition preview of the selected node is automatically displayed in the Preview tab of the window panel at the bottom of the main window.

> **Tip:** To view the YANG definition of the entire module in a new window, select the ***File / Export Preview*** command.

7. To add additional nodes to the module and set their properties, repeat the steps 2 – 5 above. You can use the pop-up menu (***Add*** command) or the drag&drop technique to add additional nodes to the YANG tree.

> **Drag&Drop tips:**
>
> To **move** a node (and all its child nodes) in the YANG tree, **drag&drop** the node from its current position to the desired position in the tree hierarchy.
>
> To **copy** a node (and all its child nodes), select the source node and **drag&drop** the node from its current position to the desired position while holding down the **CTRL** keyboard key (**ALT/Option** key on Mac OS X).

8. By default, when adding some node types (e.g., **leaf**, **leaf-list**, etc.) the Error or Warning validation message appears in the **Errors tab** of the Output window and a small **error** ( ) or **warning** ( ) overlay icon appears on the given node indicating the node has an invalid or inconsistent setting (Figure 17). The error/warning icon is propagated to parent nodes all the way up to the root node in the YANG tree. The validation messages and icons that identify the invalid node is displayed because the real-time YANG validation is enabled by default. The message and icon will automatically disappear when you edit the node properties (e.g., set the Type property of the leaf or leaf-list node) or otherwise fix the inconsistency reported by the error or warning message (Figure 18). For more information about YANG validation and validation messages, please refer to the Validate Modules section of this manual.

> **Tip:** To disable the real-time validation, open the **Preferences** dialog box (***Edit/Preferences***), select the ***Design Mode*** entry in the navigation tree and uncheck the ***Realtime validation*** checkbox in the Design Mode Preferences panel. You can then still validate the module and submodules manually by using the ***Module/Validate*** command.

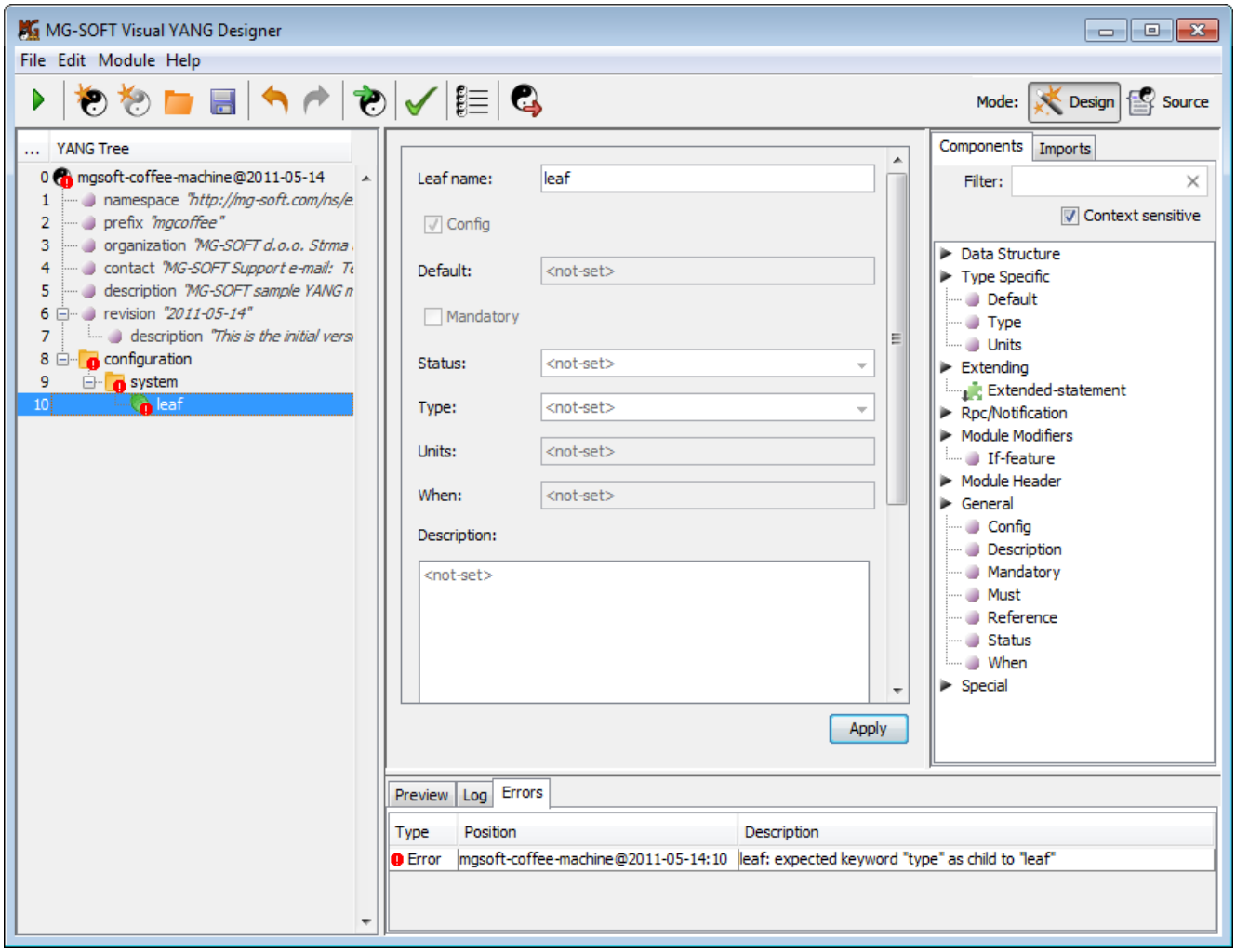


Figure 17: A newly added leaf node and the error indication

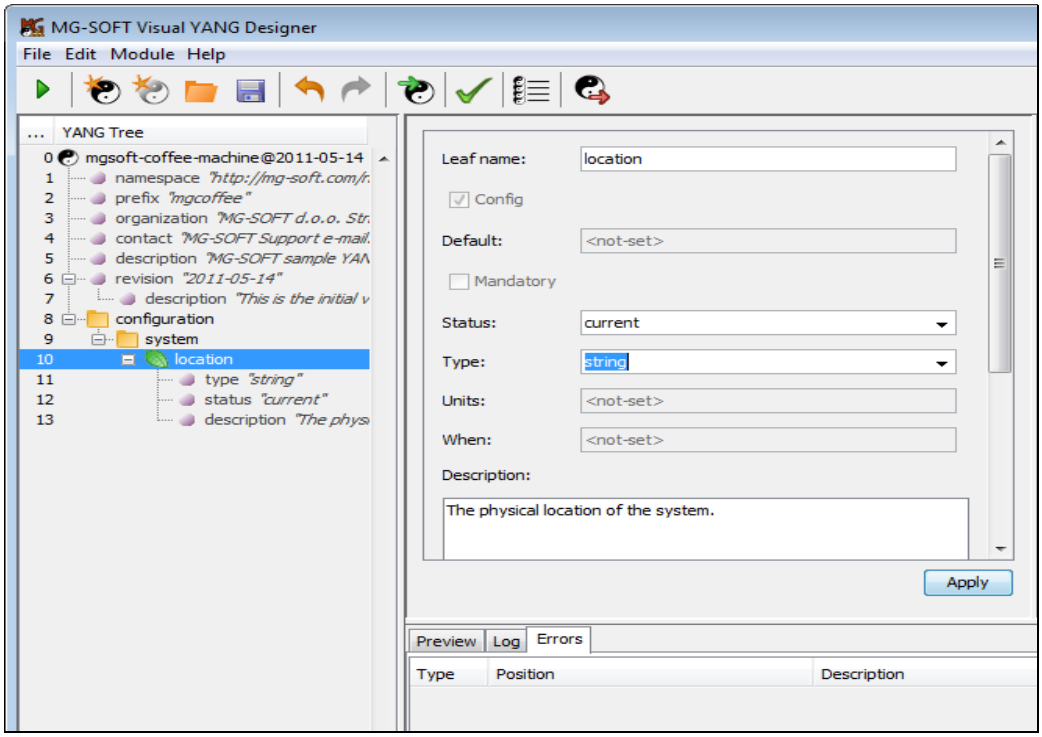


Figure 18: The error message and icon disappears after setting the Type property of the leaf node

5.2 Importing Modules

This section describes how to import a YANG module and reference its definitions in the current module or copy nodes from the imported modules to the current module. All described operations refer to the Design mode of operation.

Visual YANG Designer can import files written in the native YANG format, in XML-based YIN format, and in YANG Designer Document format (.ydd).

1. Select the ***File / Import*** command from the main menu.

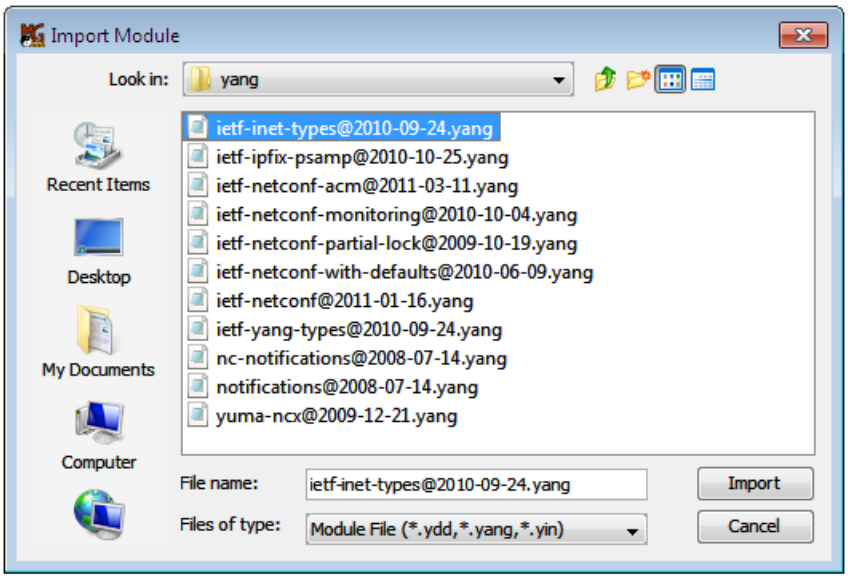


Figure 19: Importing a YANG module

2. The Import Module dialog box appears (Figure 19).
   - If the file containing the YANG module you want to import does not have the `.yang`, `.yin` or `.ydd` filename extension, select the ***All Files*** file mask from the ***Files of Type*** drop-down list in the Import Module dialog box.
   - Navigate to the folder containing YANG or YIN or YANG Designer Document files, select the desired file from the list and double-click it or click the ***Import*** button.

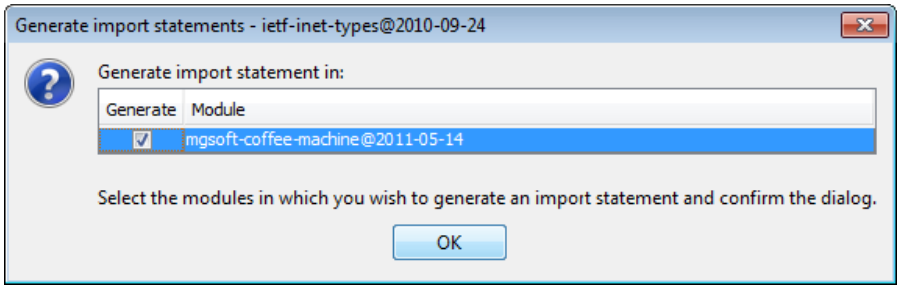

Generate import statements - ietf-inet-types@2010-09-24
Generate import statement in:
Generate
Module
mgsoft-coffee-machine@2011-05-14
Select the modules in which you wish to generate an import statement and confirm the dialog.
OK

Figure 20: Generate Import Statements dialog box

   - The Generate Import Statements dialog box appears (Figure 20), which prompts you to select the module or submodule to which the import statements (properties) will be added.
   - In the Generate Import Statements dialog box, check the checkbox in front of the module or submodule you are importing the module into and click the ***OK*** button to close the dialog and add import properties sub-nodes to the selected module/submodule.
3. This imports the YANG module from the selected file and display its tree structure in the Components window panel - **Imports** tab (Figure 21). If the module tree structure is not shown, right-click the module node and choose the ***Expand*** command from the mouse context menu to expand the schema tree of the module. Note that the import property is added to the module node in the YANG Tree window panel (Figure 21). This import property is automatically configured to import the module displayed in the Components panel – Imports tab (i.e., using the prefix of the imported module).
4. Notice the small warning icon displayed on the **import** sub-node ( ) and the Warning message in the Errors tab of the Output panel, stating that the imported module is not used (Figure 21). This warning messages and icon identified the offending node is displayed because the real-time YANG validation is enabled by default. The message and icon will disappear when you use a statement from the imported module in the (sub)module you are designing.

> **Tip:** To disable the real-time validation, open the **Preferences** dialog box (***Edit/Preferences***), select the ***Design Mode*** entry in the navigation tree and uncheck the ***Realtime validation*** checkbox in the Design Mode Preferences panel.

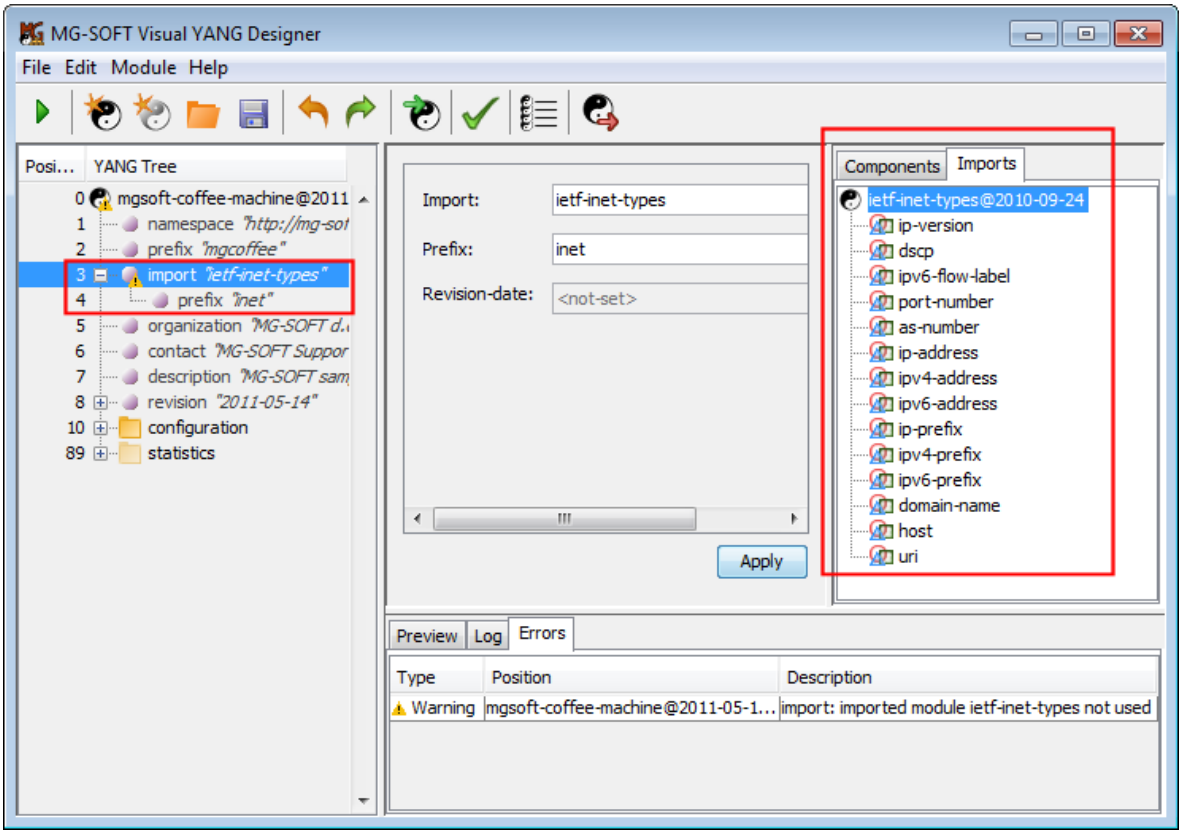


Figure 21: Imported YANG module displayed in the Components panel - Imports tab

5. Thereafter, the statements in the imported module may be referenced in the current module, for example, nodes in the current module can use the **typedefs** defined in the imported module. In addition, nodes from the imported module can be copied to the current module/submodule by using the drag&drop technique.

6. To view the properties of any node in the imported module, right-click the node, whose properties you want to view, and choose ***YANG Node Properties*** command from the pop-up menu (Figure 22).

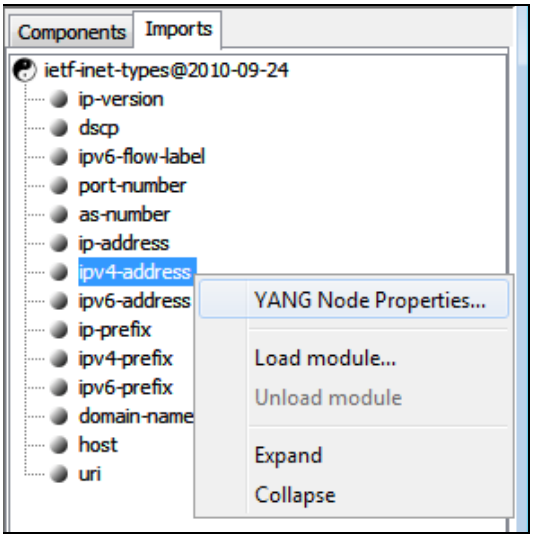


Figure 22: Selecting the option to view the properties of a node in imported module

7. The YANG Node Properties window opens (Figure 23) displaying all the properties of the selected node as defined in the imported YANG definition file.

> **Tip:** Check the ***Always on top*** checkbox in the Yang Node Properties window to keep this window in the foreground, automatically updating its contents while you click other nodes in the Imports tab.

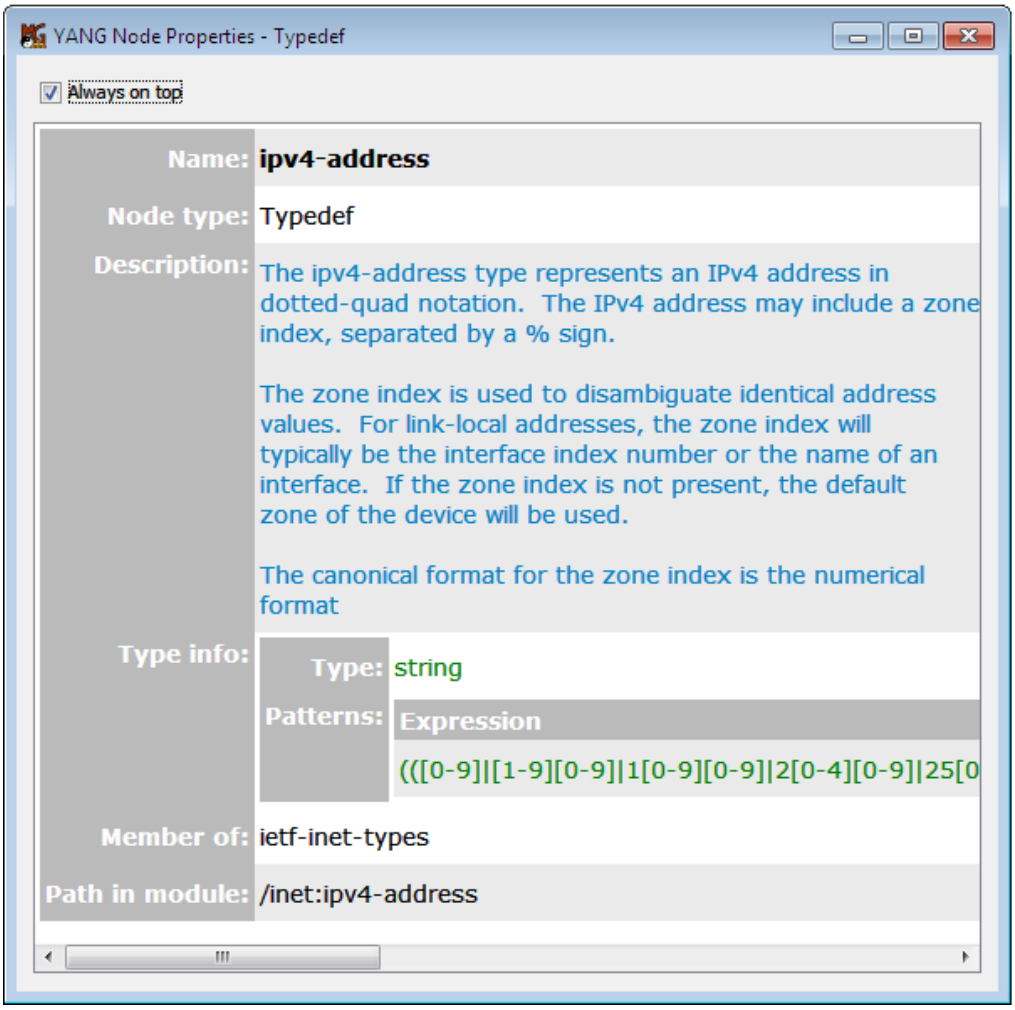

YANG Node Properties - Typedef
Always on top
Name: ipv4-address
Node type: Typedef
Description: The ipv4-address type represents an IPv4 address in dotted-quad notation. The IPv4 address may include a zone index, separated by a % sign.
The zone index is used to disambiguate identical address values. For link-local addresses, the zone index will typically be the interface index number or the name of an interface. If the zone index is not present, the default zone of the device will be used.
The canonical format for the zone index is the numerical format
Type info:
Type: string
Patterns:
Expression
(([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0
Member of: ietf-inet-types
Path in module: /inet:ipv4-address

Figure 23: Viewing properties of the selected node as defined in imported module

5.2.1 Using Nodes from Imported Modules

Nodes from the imported modules can be either used (referenced by) or copied into the current module in the YANG Tree window panel. This section describes how to use the drag&drop technique to import certain types of nodes and reference them in the current module.

Importing is allowed only for specific top-level YANG nodes (typedef, grouping, identity, extension, feature and module), where the dragged node type automatically changes when dropped, according to the following table:

Source (dragged) YANG node type	Destination (dropped) YANG node type
Grouping	Uses
Identity	Base
Extension	Extended statement
Typedef	Type
Feature	If-feature
Module	Import

The following section describes how the **Typedef** and **grouping** nodes defined in imported module can be used in the current module/submodule.

1. To use a **typedef** defined in the imported module, select the target node in the module you are designing (YANG Tree panel), choose the desired **typedef** from the ***Type*** drop-down list in the Node Properties panel (Figure 24) and click the ***Apply*** button to apply the changes. The imported types are listed using the `prefix:typdefname` notation (e.g., `inet:ipv4-address`).

> **Note:** When you import a module to the Components panel – Imports tab, the ***Type*** property drop-down list in the current module is automatically extended with all type definitions (typedefs) from the imported module (in addition to the base types and all typedefs from the current module). When selecting imported types from the ***Type*** drop-down menu, description of the respective type, as specified in the imported module, is automatically displayed in a tooltip – see Figure 24.

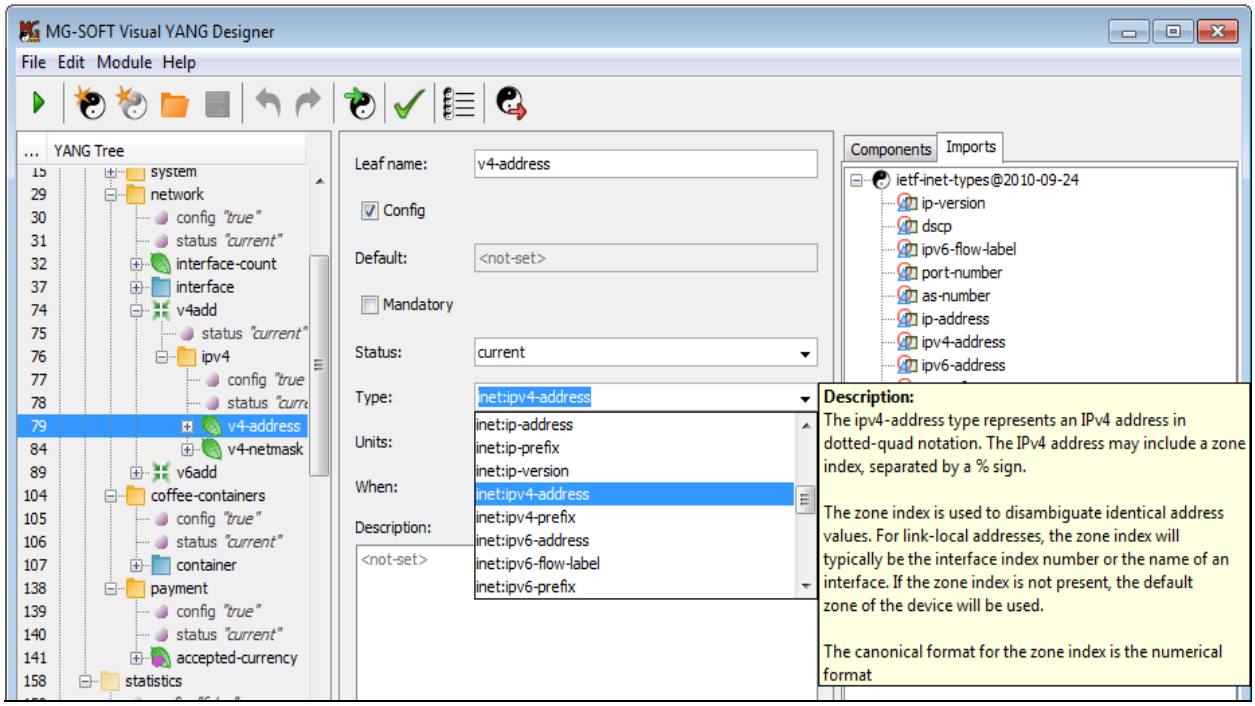


Figure 24: Selecting a type defined in the imported module (e.g., inet:ipv4-address)

2. To use a **grouping** type of node from the imported module, select the node you want to use in the Components panel, Imports tab and drag&drop it to the appropriate position in the current module in the YANG Tree panel (Figure 25).

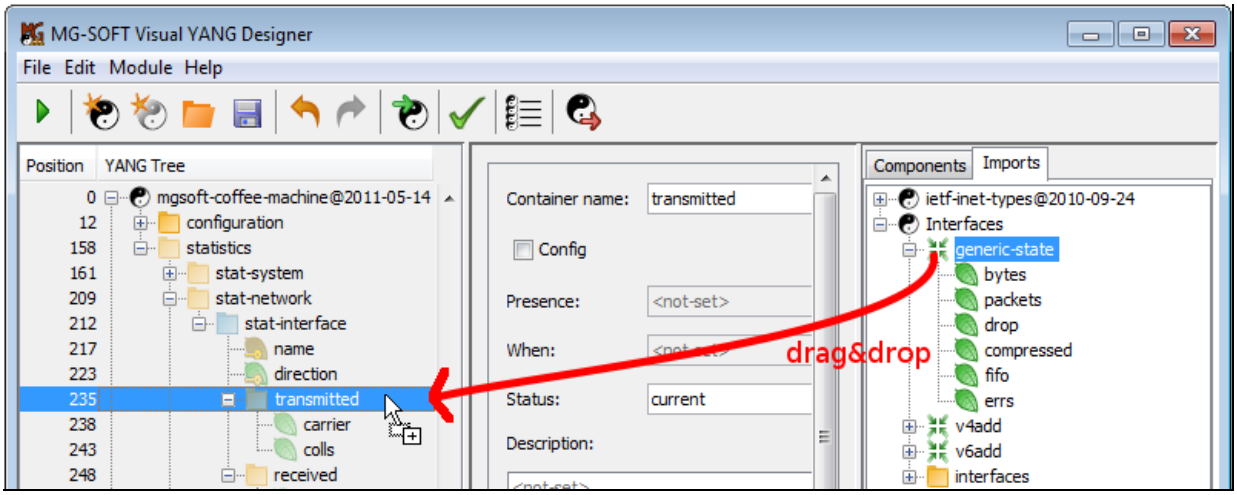


Figure 25: Adding a grouping node from the imported module (drag&drop)

3. The dropped node is added to the desired position in the YANG tree and properly references the node from the imported module. For example, when you drop a **grouping** node, the **uses** node that uses this grouping is automatically created on the selected position in the hierarchy of the current module (Figure 26).

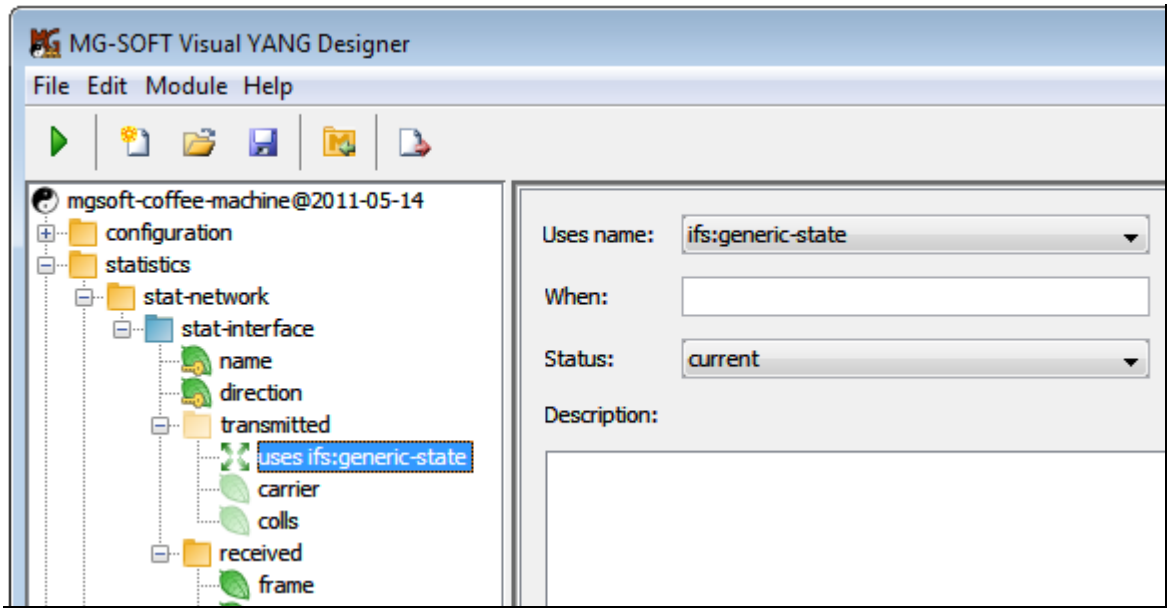


Figure 26: A uses node with imported grouping

5.2.2 Copying Nodes from Imported Modules

Nodes from the imported modules can be copied to the current module/submodule by using the drag&drop technique while pressing the **CTRL** key (**ALT/Option** key on Mac OS), as described in this section.

1. To create a copy of a node or a subtree defined in the imported module, select the desired node in the Components panel, Imports tab, press and hold down the **CTRL** keyboard key (**ALT/Option** key on Mac OS) and **drag&drop** it to the appropriate position in the current module or submodule in the YANG Tree panel (Figure 25).

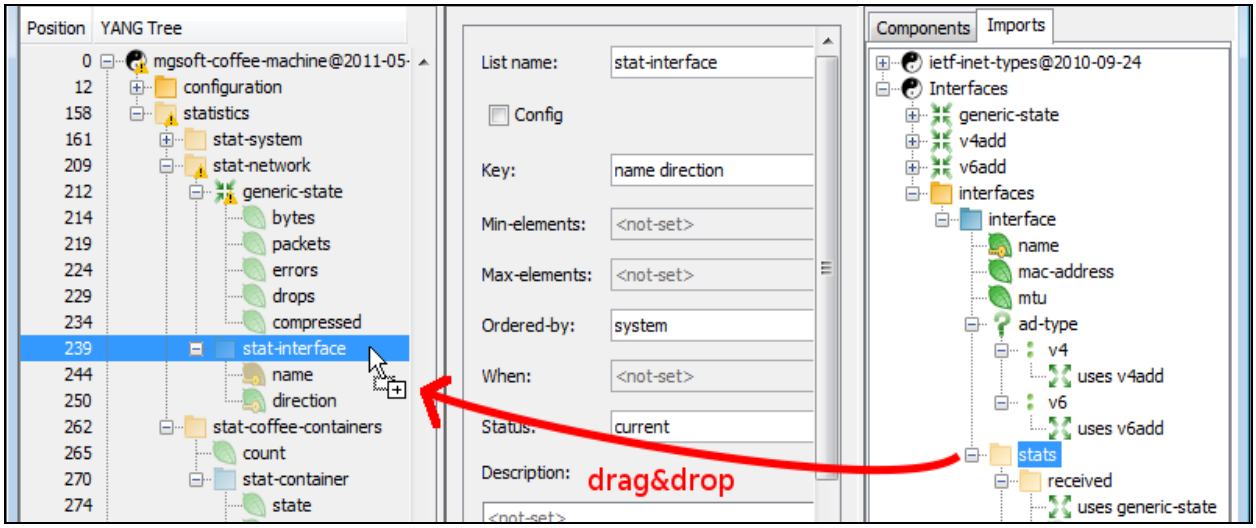


Figure 27: Copying a container node from the imported module (drag&drop+CTRL)

2. A copy of a dropped node with all its child nodes is added to the selected position in the YANG tree (Figure 28). The properties of the copied objects are the same as in the source module. You can use copied nodes or subtrees as a base for defining similar objects in the current module/submodule.

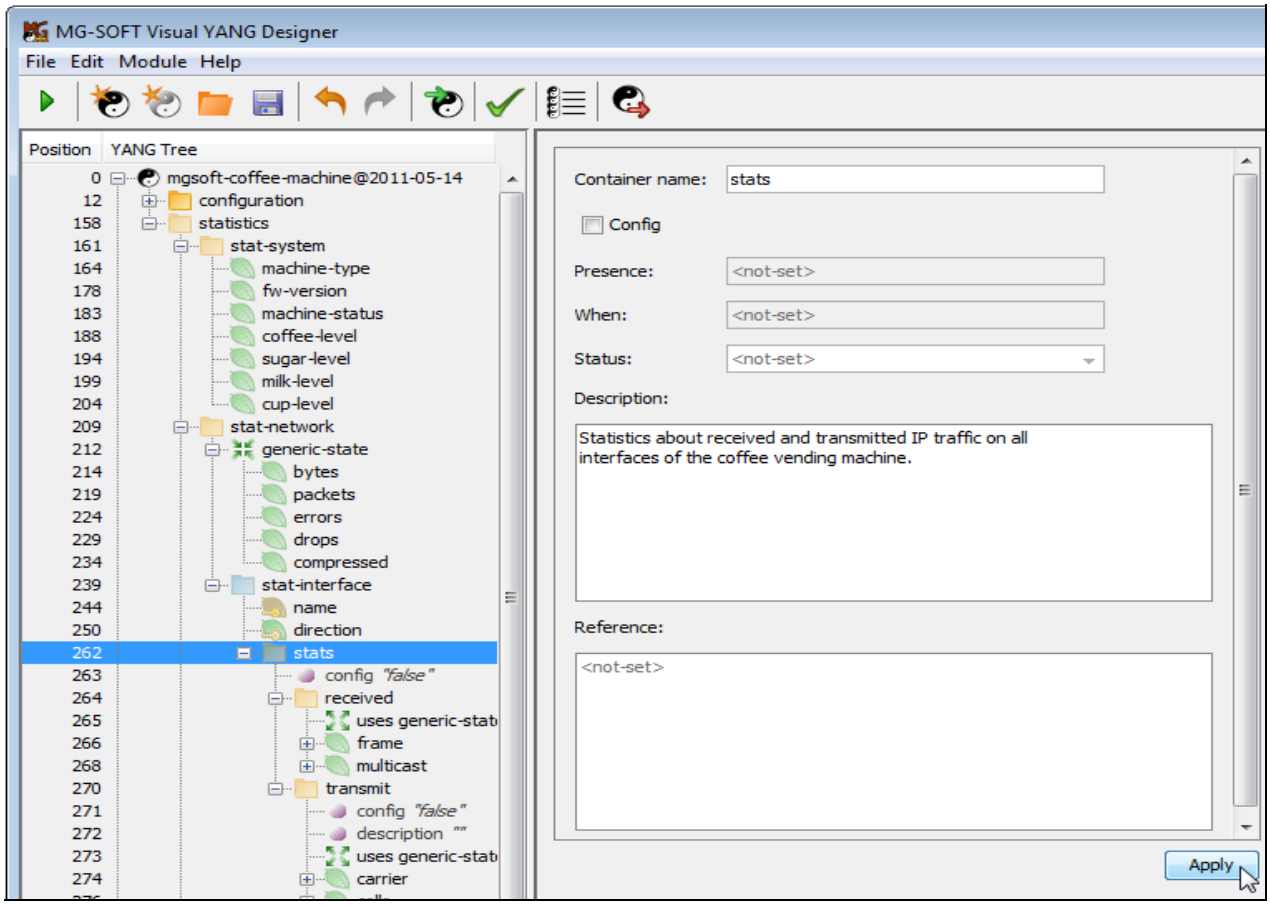


Figure 28: A copy of a subtree from imported module is added to the current module

3. Edit the properties of the copied nodes to match your needs and apply the changes by clicking the ***Apply*** button for each node in the Node Properties window panel (Figure 28).

5.3 Creating a Submodule

YANG structures data models into modules and submodules. While the primary unit in YANG is a module, a YANG module can itself be constructed out of several submodules. Submodules allow a module designer to split a complex model into several pieces where all the submodules contribute to a single namespace, which is defined by the module that includes the submodules.

Submodule is a partial module definition that contributes derived types, groupings, data nodes, RPCs, and notifications to a module. A YANG module can be constructed from a number of submodules.

A module uses the "include" statement to include its submodules, and the "import" statement to reference external modules. Similarly, a submodule uses the "import" statement to reference other modules, and uses the "include" statement to reference other submodules within its module.

1. In the main window, choose the ***File / New / Submodule*** command.
2. The New Submodule dialog box appears (Figure 29), which prompts you to enter the basic properties for the new submodule.

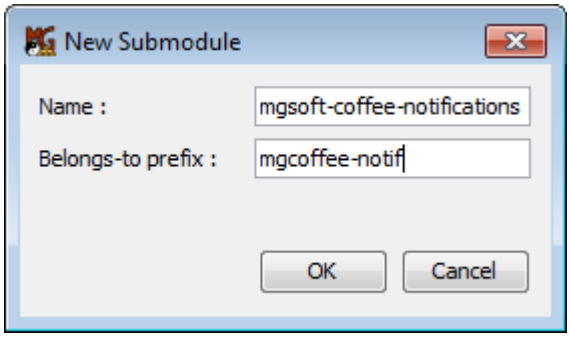


Figure 29: Specifying the basic parameters for the new submodule

3. Into the ***Name*** input line, enter a name for the new submodule you are creating.

> Private submodule names are assigned by the organization owning the submodule without a central registry. It is recommended to choose submodule names that will have a low probability of colliding with standard or other enterprise modules and submodules, e.g., by using the enterprise or organization name as a prefix for the submodule name. The submodule name follows the rules for identifiers.

4. Into the ***Belongs-to prefix*** input line, enter a prefix string for this submodule. All definitions in the local submodule and any included submodules can be accessed by using this prefix. Note that the new submodule you are creating automatically belongs to the module currently displayed in the YANG Tree panel.
5. After you have specified the above settings, click the ***OK*** button to create the new submodule and close the New Submodule dialog box. A new node of type **submodule** ( ) is added to the YANG Tree window panel and the main properties of the submodule node are displayed as its sub-nodes ( ). The newly added submodule node is automatically selected and its main properties are displayed also in the Node Properties window panel in the center of the main window (Figure 30).

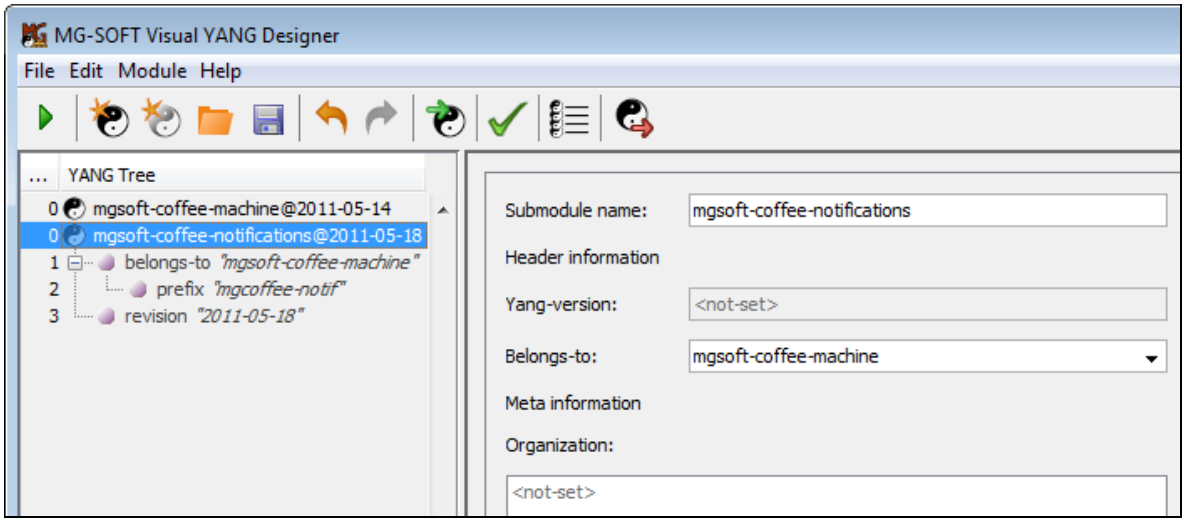


Figure 30: A newly added submodule node and its properties

6. To configure the remaining (optional) properties of the submodule node (e.g., Organization, Contact, Description, Reference, Revision, etc.), you have two options:

> For the exact meaning of the remaining module node properties (substatements), please refer to the YANG specification, section 7.2.

- In the Node Properties window panel, set the remaining properties by entering the text into the corresponding input fields (e.g., **Organization**, **Contact**, **Description**, **Reference**, **Revision**, etc.) and click the ***Apply*** button at the bottom of the Node Properties window panel to apply the changes.
- Alternatively, in the Components panel, select the property sub-node ( ) you want to configure (e.g., **Organization**, **Contact**, **Description**, **Reference**, **Revision**, etc.) and drag&drop it to the submodule node in the YANG Tree panel. Then, configure its settings that are displayed in the Node Properties window and click the ***Apply*** button in the Node Properties panel to apply the changes (Figure 31).

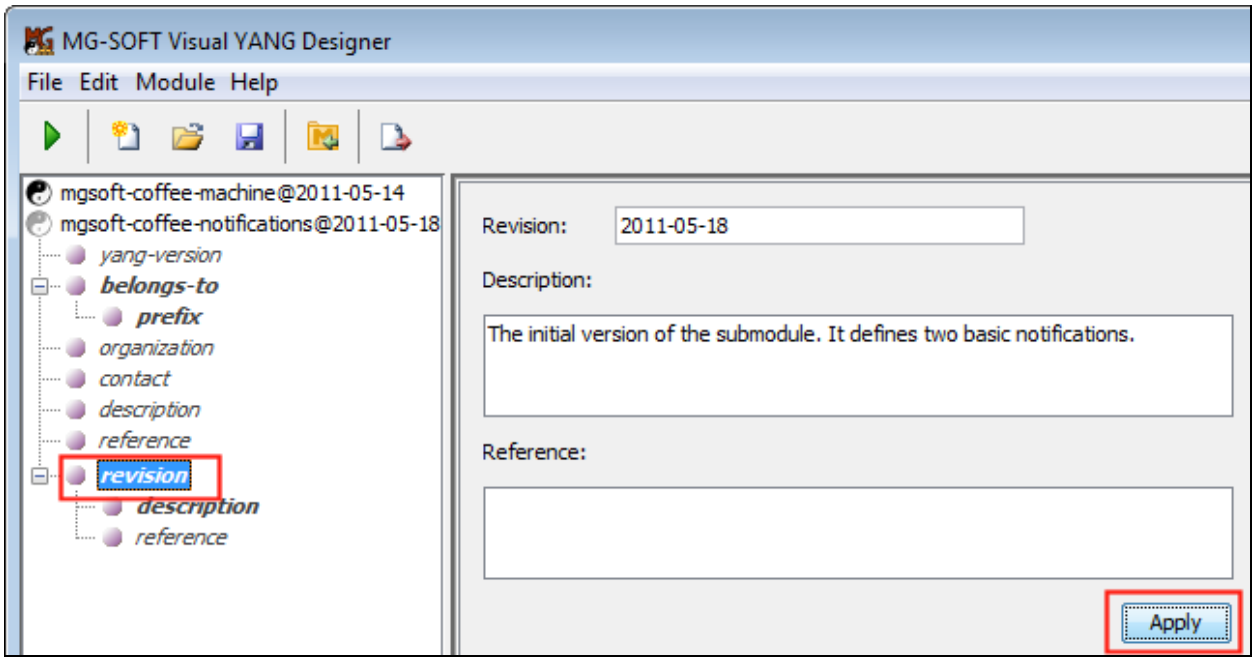


Figure 31: Setting the revision property of a submodule node

Note: When a submodule is created, Visual YANG Designer automatically adds the proper "include" statement to the main module to include the given submodule.

5.3.1 Adding Nodes to a Submodule

Submodule is a partial module definition that contributes derived types, groupings, data nodes, RPCs, and notifications to a module. The method of designing the data model in a submodule (i.e., adding nodes to a submodule and setting their properties) is the same as for a module. Please refer to the Adding Nodes to a New Module section for instructions on adding nodes and setting their properties.

5.4 Saving Modules

You can save the module you are currently designing to MG-SOFT YANG Designer document format (*.ydd) at any time. This is MG-SOFT proprietary binary file format for storing YANG Designer projects. If the YANG Tree panel contains also submodules, these will be saved together with the module into the same YANG Designer document (*.ydd) file. Once saved, the YANG Designer document file can be reopened in MG-SOFT Visual YANG Designer (using the ***File / Open*** command) so that you can edit/update the module and submodules (if present) at any later time.

1. To save the current module and all submodules (if any) to a file, select the ***File / Save*** command. The standard Save As dialog box appears.
2. In the Save As dialog box, navigate to the destination, where you want to save the file to and specify the file name into the ***File name*** input line.
3. Click the ***Save*** button to close the dialog box and write the module to the YANG Designer file format (*.ydd).

Tip: Files that have been recently opened in Visual YANG Designer, can be quickly re-opened by selecting their names from the list of recent files (***File / Recent Files***)

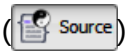
6 DESIGN NEW YANG MODULE IN SOURCE MODE

In addition to the visual design mode, the software provides also an advanced source mode of operation. The Source mode lets you design NETCONF YANG modules and submodules by writing the YANG code directly in the fully-fledged YANG Source Editor, which incorporates effective YANG coding mechanism, like intelligent code completion, syntax coloring, find usages, go to definitions, find/replace, etc.

The Source mode is aimed at more experienced YANG authors who prefer writing YANG code themselves.

This section describes the basic principles of designing YANG modules in the Source mode.

6.1 Switching to Source Mode

1. To activate the Source mode, select the **View / Source Mode** from the main menu or click the ***Source mode*** toolbar button (

).
2. The Visual YANG Designer Source mode desktop appears with the YANG Source Editor panel occupying the central section.

6.2 Creating a New Module in Source Mode

The procedure of creating a new module and submodule in Source mode is the same as in Design mode.

For detailed instructions on creating a new module, please refer to steps 1-6 in the section Creating a New Module.

After completing the steps 1-6 described in section Creating a New Module, the new **module** statement with the basic substatements (**namespace, prefix and revision**) is inserted into the YANG Source Editor panel (Figure 32). The YANG Tree Preview window panel on the left displays the structure of the new YANG module in a graphical manner (similarly as the YANG Tree panel in Design mode).

You can edit the module further in the YANG Source Editor panel, as described in the next section.

> **Tip:** For detailed instructions on creating a new **submodule**, please refer to steps 1-5 in the section Creating a Submodule.

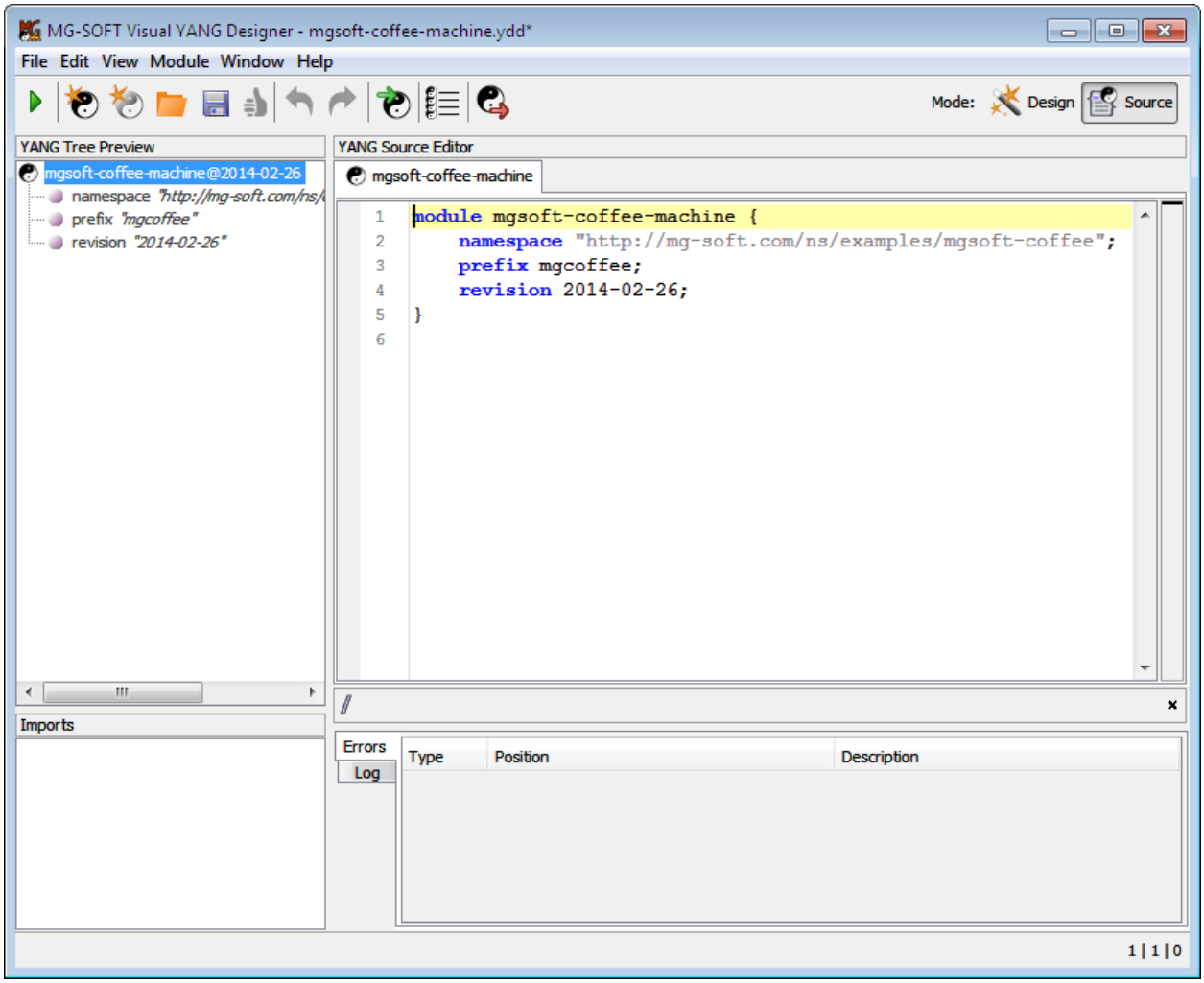


Figure 32: A newly created *module* statement in the YANG Source Editor panel

6.3 Using YANG Source Editor

YANG Source Editor panel displays and lets you edit the source code of the YANG module and submodules you are designing. YANG Editor incorporates the YANG syntax coloring, YANG auto-complete feature, Find Usages, Go to Definition functionality, etc. as described in this section.

6.3.1 Syntax Coloring and Pretty Print Formatting

YANG Source Editor automatically recognizes all YANG keywords, arguments and punctuation marks in the module and applies YANG syntax coloring and pretty print formatting accordingly. By default, all keywords are displayed in blue, arguments in black, descriptions in gray and comments in green letters. You can adjust the colors to your taste in the **Preferences** dialog box (**Edit/Preferences**), on the **Source Mode** panel.

Pretty print formatting automatically indents YANG statements and substatements and properly aligns the braces of nested (sub)statements. To apply pretty print formatting to the YANG code that is not pretty formatted (e.g., code originating from other sources), select the ***Edit / Format Source*** command from the menu.

6.3.2 Writing YANG Code and Using Auto-Complete Feature

Visual YANG designer incorporates an advanced YANG code auto-complete feature that suggests the completion of keywords (e.g., leaf, leaf-list, container, list, type, etc.) and arguments that have some defined values (e.g., type, status, config, mandatory, key, uses, extension, if-feature, etc.). Furthermore, a description of each completion entry is displayed in a tooltip next to the auto-complete drop-down list for better understanding of available options.

1. To activate the auto-complete feature, place your cursor to the position in the document where you want to write a YANG statement, enter zero or more characters and press the **CTRL+SPACE** keys.
2. The auto-complete drop-down list appears, containing all possible keywords or arguments (depending on the current cursor position in the code) that begin with the entered characters (if any). A description of each completion entry is shown in the tooltip (Figure 33).

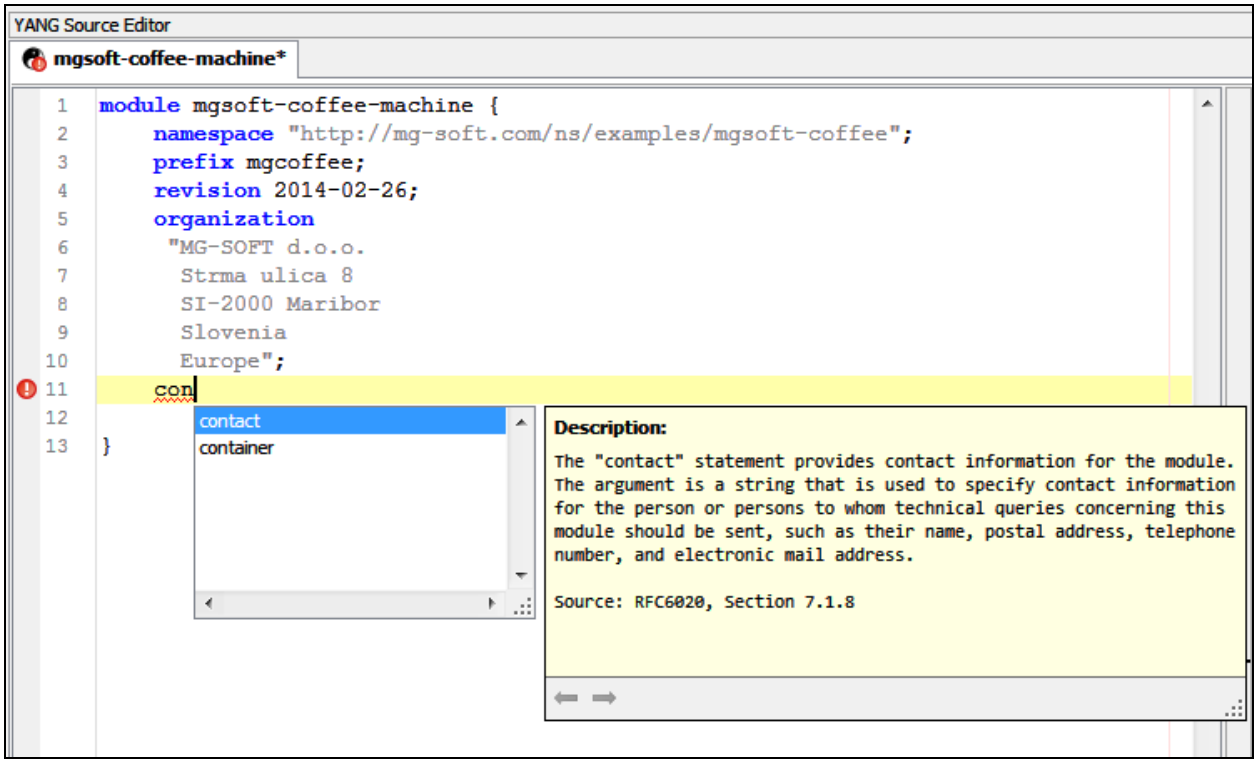

YANG Source Editor
mgsoft-coffee-machine*
module mgsoft-coffee-machine {
namespace "http://mg-soft.com/ns/examples/mgsoft-coffee";
prefix mgcoffee;
revision 2014-02-26;
organization
"MG-SOFT d.o.o.
Strma ulica 8
SI-2000 Maribor
Slovenia
Europe";
con
contact
container
}
Description:
The "contact" statement provides contact information for the module. The argument is a string that is used to specify contact information for the person or persons to whom technical queries concerning this module should be sent, such as their name, postal address, telephone number, and electronic mail address.
Source: RFC6020, Section 7.1.8

Figure 33: the auto-complete drop-down list and description tooltip in the YANG Editor

3. If the list of possible completions is large, type one or more additional characters that serve as a filter that will narrow down the number of completion entries in the list.
4. Select the desired completion entry from the auto-complete drop-down list by double-clicking it or selecting it and pressing the **Enter** key.

5. The selected word is inserted to the current cursor position in the document, completing the entered phrase (e.g., “contact” instead of “con”).
6. Use the steps 1-5 again to quickly and correctly add additional statements or arguments to the code of the YANG module you are designing.

> **Tip:** You can use the auto-complete feature also to **import** known modules. To do this, type the **import** statement, a space and zero or more module name characters and press **CTRL+SPACE** to display the auto-complete drop-down list containing the names of known modules. In a similar way, you can include a submodule that belongs to the module you are editing.

6.3.3 Using Undo/Redo and Go to Last Edit Location Functions

While you edit the code in the YANG Source Editor panel, the software automatically maintains the history of changes made in the YANG module and all of its submodules. This enables you to use the **Undo** and **Redo** functionality to undo and redo past edits or to simply go **back** to the locations of past edits without modifying the document.

Using the Undo/Redo feature:

1. To undo the last change in YANG code of the currently selected module or submodule, select the ***Edit / Undo*** command or press the ***CTRL+Z*** keyboard keys.
2. The software reverses the last change in the code. To undo more changes, select the ***Edit / Undo*** command or press the ***CTRL+Z*** (***⌘+Z*** on Mac) keys repeatedly.
3. If you want to re-apply a change that has been undone, select the ***Edit / Redo*** command or press the ***CTRL+Y*** (**SHIFT+⌘+*Z*** on Mac) keyboard keys.

> **Note:** **Undo** and **Redo** commands apply only to the document in the currently selected tab of the YANG Source Editor panel. Select a different tab if you want to undo or redo changes in the document displayed there.

Using the Go To Last Edit Location feature:

1. To go to the location in YANG code that you have edited last, select the ***Edit / Last Edit Location*** command or press the ***CTRL+Q*** keyboard keys.
2. The software moves the cursor and focus to the location in YANG code that you have edited last. This can be either in the module or any of the submodules, depending on where the code has been edited last.
3. To go further back in history of edited locations, select the ***Edit / Last Edit Location*** command again or press the ***CTRL+Q*** keyboard keys again.
4. To stop browsing the past edited locations and go back to the location where you edited the code before visiting past locations, click a white space in the current document and select the ***Edit / Last Edit Location*** command again (or press the ***CTRL+Q*** keys).

6.3.4 Committing Changes

To apply the changes you have made in the Source mode to all parts of the application (i.e., to Source mode, Design mode and Export module), you need to **commit** the changes. Without committing, the source code changes will be lost if you, for example, exit the application without saving or switch to Design mode and further edit the YANG module there.

1. If uncommitted source code changes exist, the name of the module in the respective YANG Source Editor tab is displayed in **bold** typeface with an asterisk (*****) appended to it (Figure 34).

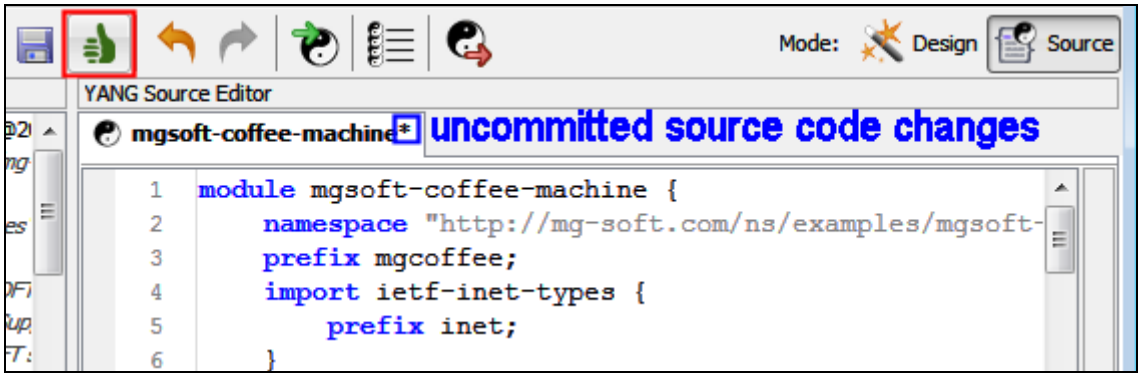


Figure 34: Committing changes made in YANG Editor

2. To commit the changes, select the ***Edit / Commit Source Changes*** menu command or click the ***Commit*** toolbar button ( ).
3. After applying the changes throughout the application, the name of the module/submodule you are editing is displayed in normal (non-bold) typeface and the asterisk (*****) mark disappears.

6.3.5 Saving Changes

It is advisable to save the document to MG-SOFT YANG Designer document format (.ydd) after making any significant change to the YANG module/submodules you are editing. Ydd is MG-SOFT proprietary binary file format for storing YANG Designer projects. If the YANG Source Editor panel contains also submodules, these will be saved together with the module into the same YANG Designer document (*.ydd) file.

1. If unsaved changes exist, an asterisk (*****) is displayed next to the document file name in the YANG Designer titlebar (Figure 35).
2. To save the current module and all submodules (if any) to the YANG Designer document file format (.ydd), select the ***File / Save*** command.

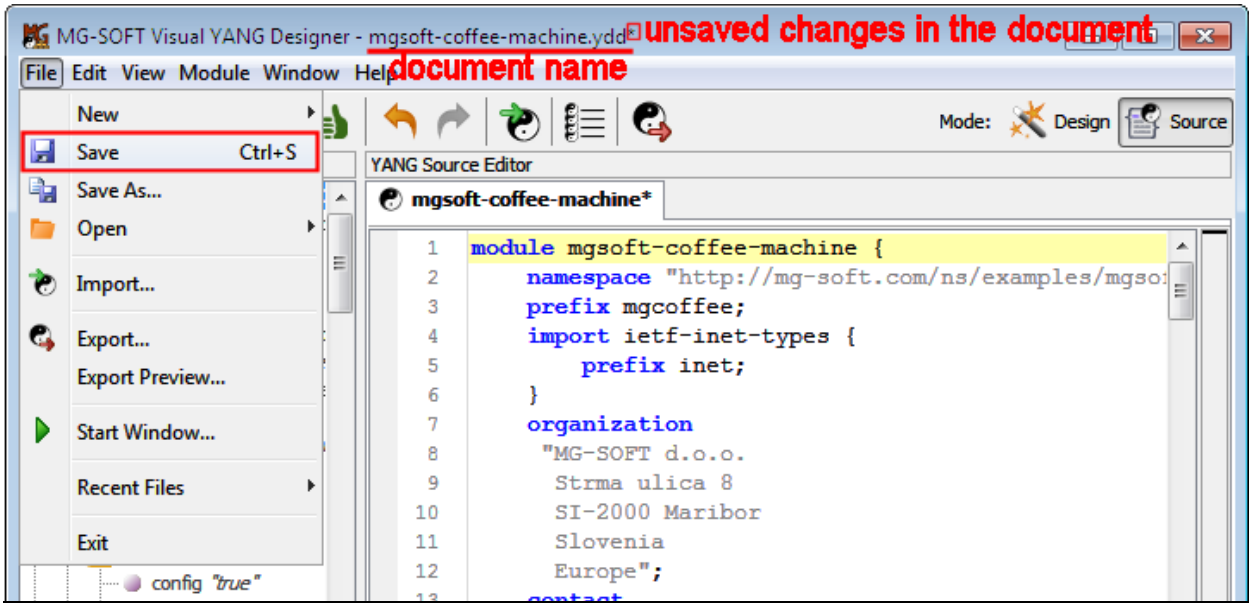


Figure 35: Saving the changes

3. If there are uncommitted changes in the source code, the **Commit and Save** dialog box appears (Figure 36).

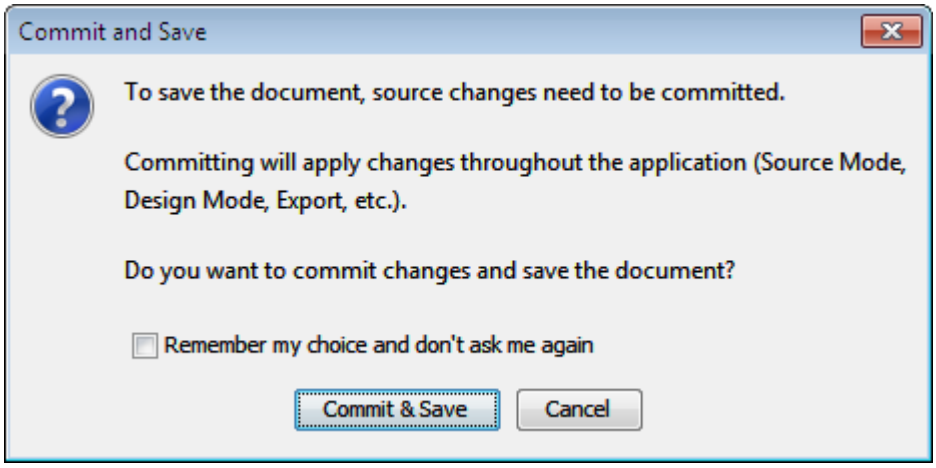

Commit and Save
To save the document, source changes need to be committed.
Committing will apply changes throughout the application (Source Mode, Design Mode, Export, etc.).
Do you want to commit changes and save the document?
Remember my choice and don't ask me again
Commit & Save
Cancel

Figure 36: Commit and Save dialog box

4. In the Commit and Save dialog box, click the ***Commit and Save*** button to allow committing the changes and saving the document to YANG Designer file format (*.ydd).

> **Tip:** If you select the ***Remember my choice an don't ask me again*** option before clicking the ***Commit and Save*** button, the software will thereafter automatically commit the changes before saving the document (without displaying this dialog box). To enable displaying the Commit and Save dialog box on Save operation again, select the ***Show Commit and Save dialog*** in the ***Preferences*** dialog box, ***Source mode*** view.

5. The standard Save As dialog box appears. In the Save As dialog box, navigate to the destination, where you want to save the file to and specify the file name into the ***File name*** input line.
6. Click the ***Save*** button to close the dialog box and write the module to the YANG Designer file format (*.ydd).

7. Once the document has been saved, you can quickly save additional changes to it by selecting the ***File / Save*** command or by using the ***CTRL+S*** keyboard shortcut.

> **Tip:** Files that have been recently opened in Visual YANG Designer, can be quickly re-opened by selecting their names from the list of recent files (***File / Recent Files***)

6.3.6 Inserting Comments

There are several ways of adding comments to the code in YANG Source Editor:

- To add a **single line comment** to the YANG code, start it with "***//***" and then write the comment. Single line comment ends at the end of the line.
- To add a **multi-line (block) comment** start it with "***/**" and end it with "***/***".

```
21      // single-line comment
22      container configuration {
23          config true;
24          status current;
25          container payment {
26              config true;
27              status current;
28              leaf-list accepted-currency {
29                  type enumeration {
30        /*  block comment start
31                          enum US$ {
32                           value 1;
33                           status current;
34                      }
35                      enum Euro {
36                           value 2;
37                           status current;
38                      }
39                      block comment end   */
40                      enum Pound {
41                           value 5;
42                           status current;
```

Figure 37: Comments in YANG Source Editor

- To **comment out** a certain section in the code, select the desired text, right–click it and choose the ***Toggle Comment*** command from the mouse pop-up menu.

  By default, comments are displayed in green letters in the YANG Source Editor panel.

> **Tip:** You can adjust the colors to your taste in the **Preferences** dialog box (**Edit/Preferences**), on the **Source Mode** panel.

6.3.7 Using Find Usages Feature

YANG Source Editor incorporates the **Find Usages** function that lets you select a **reusable definition** in the YANG code and quickly find all the places in the current module and submodules where it is being used. Reusable definitions are the local **grouping**, **typedef**, **extension**, **feature** and **identity** statements.

1. In YANG Source Editor, select a reusable definition, e.g., a grouping statement, right-click it and choose the **Find Usages** command from the pop-up menu (Figure 38).

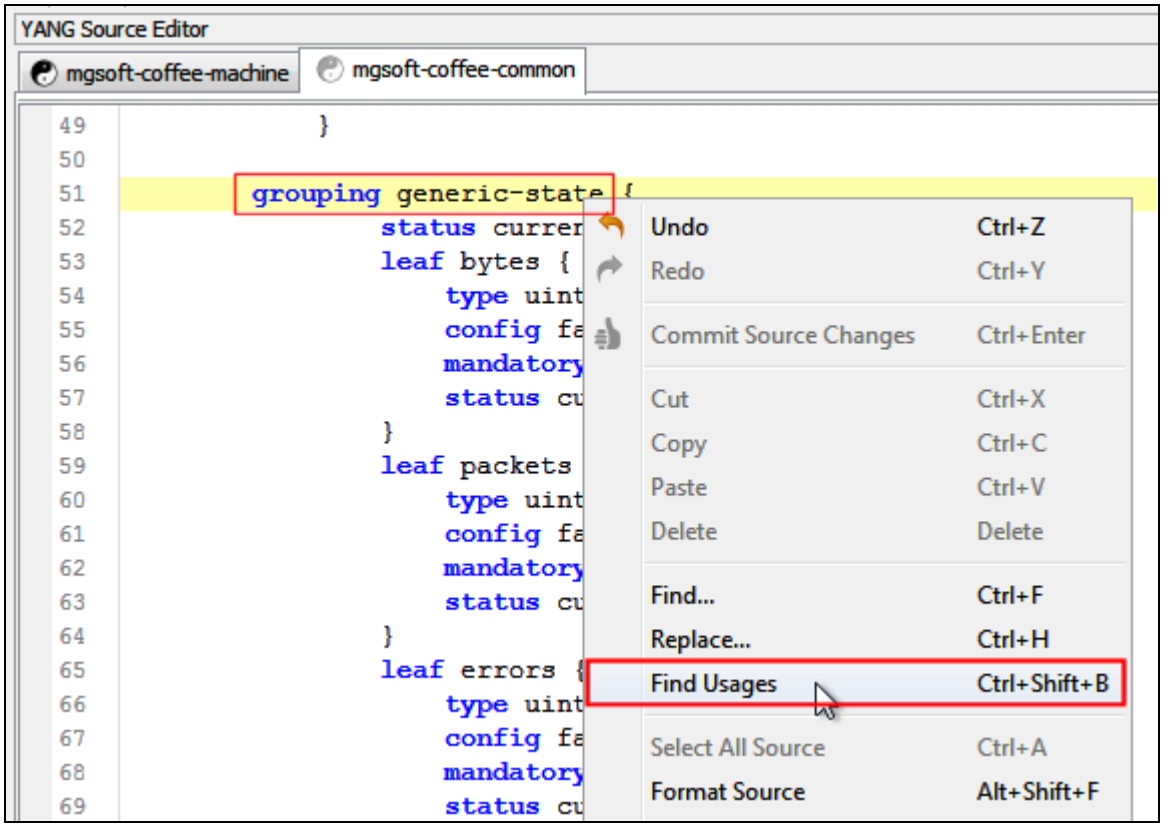


Figure 38: Selecting the *Find Usages* command

2. A new **Usages** tab appears in the Output panel at the lower-right section of the YANG Designer/Source mode. The Usages tab lists all occurrences of the reusable definition in the current module and submodules (Figure 39)
3. Double-click a usage entry in the **Usages** tab to quickly display the section of the (sub)module in the YANG Source Editor, where the reusable definition is being used (Figure 39).
4. To quickly move the cursor to the location in YANG code that you have edited last, select the ***Edit / Last Edit Location*** command or press the ***CTRL+Q*** keyboard keys.

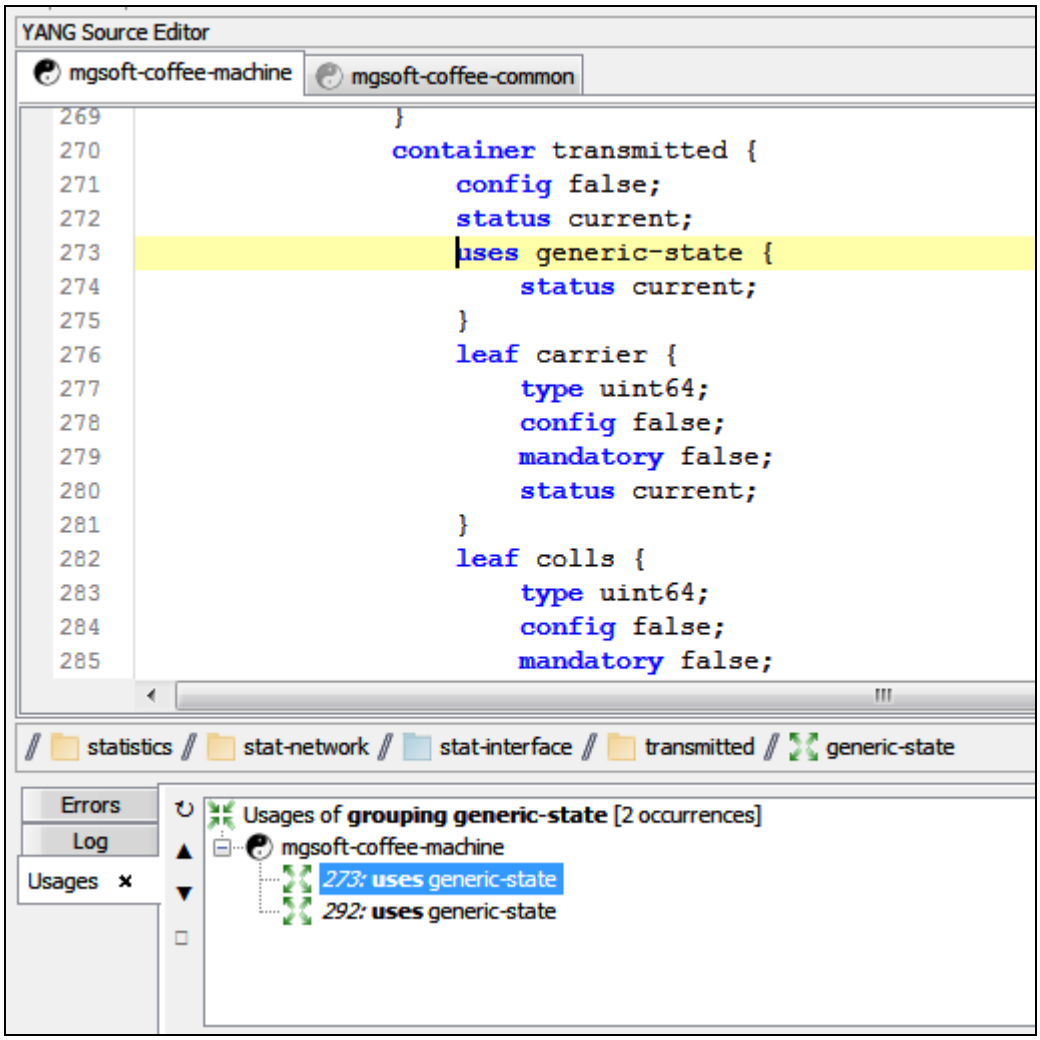


Figure 39: Inspecting the usage of the selected reusable code element

6.3.8 Using Go to Definition Feature

YANG Source Editor incorporates the **Go to Definition** function that lets you select a **reused definition** and quickly find the location in the YANG code where it is defined. Reused definitions are arguments of the **uses**, **augments**, **deviations**, **if-feature**, **base** and **type (derived)** statements, and the keywords of **extended** statements. The **Go to Definition** can be seen as a reverse function of the **Find Usages** function.

This section describes the basic principle of using the **Go to Definition** functionality in Visual YANG Designer.

1. In YANG Source Editor, select a reused definition, e.g., an argument of a **uses** statement, right-click it and choose the ***Go to Definition*** command from the pop-up menu (Figure 40).

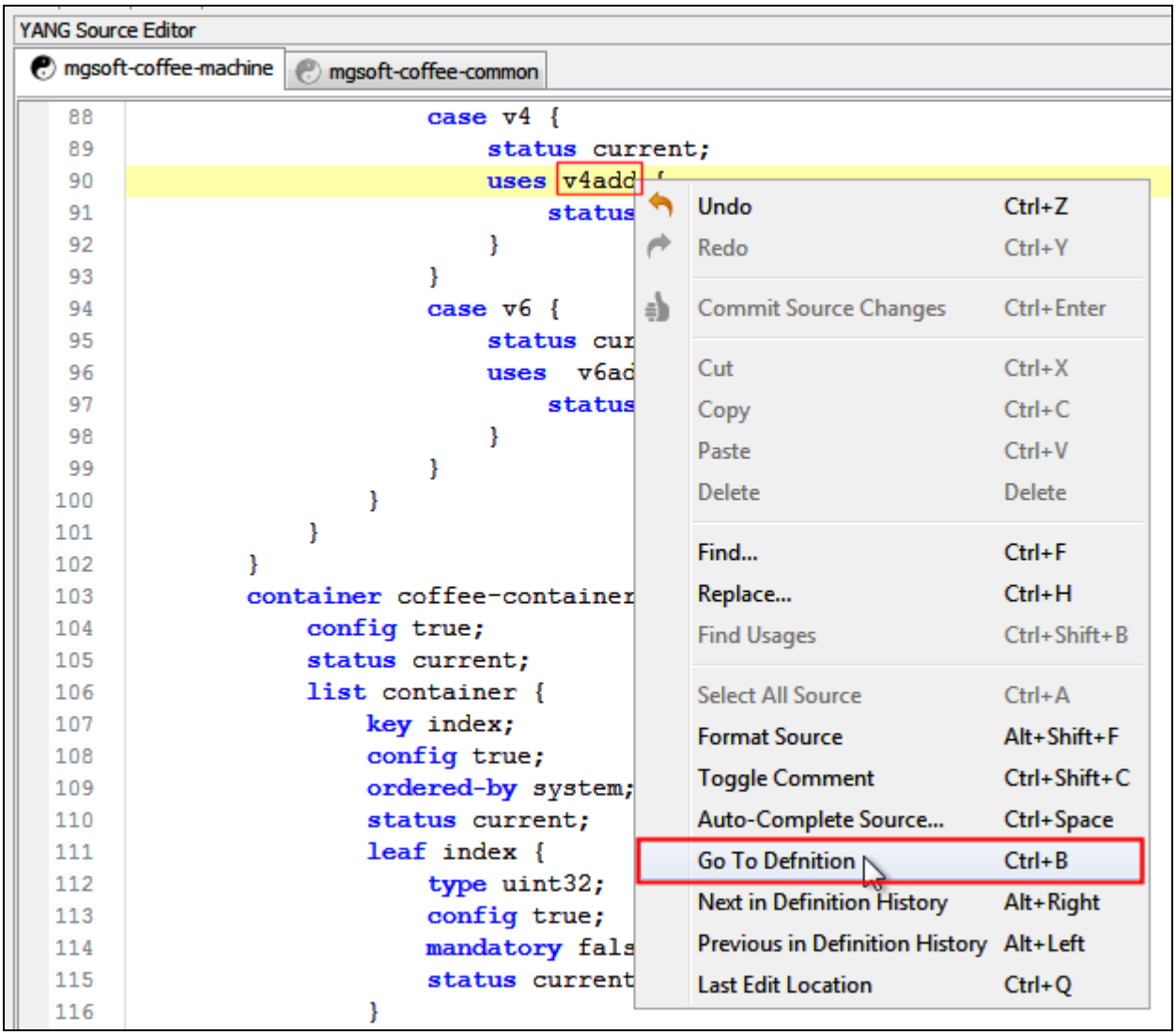


Figure 40: Selecting the *Go to Definition* command

2. YANG Source Editor displays the YANG code section where the selected reused definition is defined (Figure 41).

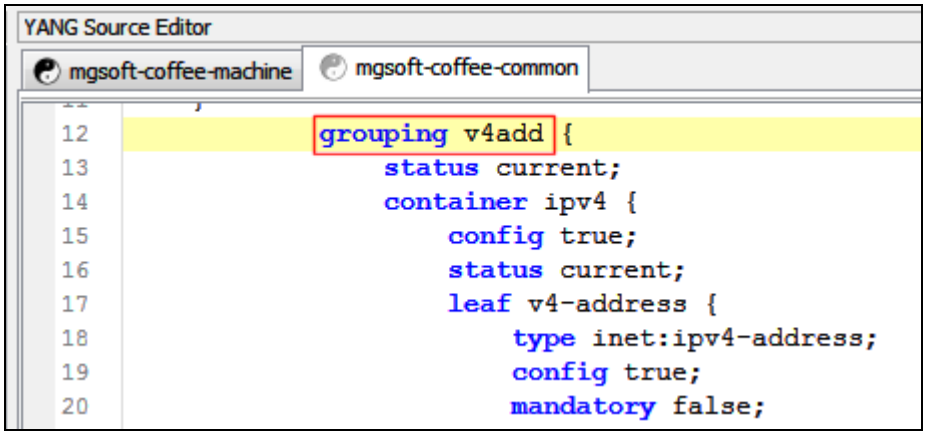


Figure 41: Viewing the referenced reusable definition in YANG Source Editor

3. To quickly navigate back to the location in YANG code from where you selected the ***Go to Definition*** command, select the ***Edit / Back*** command or press the ***ALT+←*** keyboard keys.

6.3.9 Using the Find/Replace Text Feature

This section describes how to use the convenient find and replace text feature available in Visual YANG Designer Source mode. The entire document in the given YANG Source Editor tab will be searched by default (when the "Wrap around" search option is enabled).

1. Select the ***Edit/Replace*** command or press the ***CTRL+H*** keyboard keys to display the Find/Replace toolbar (Figure 42).

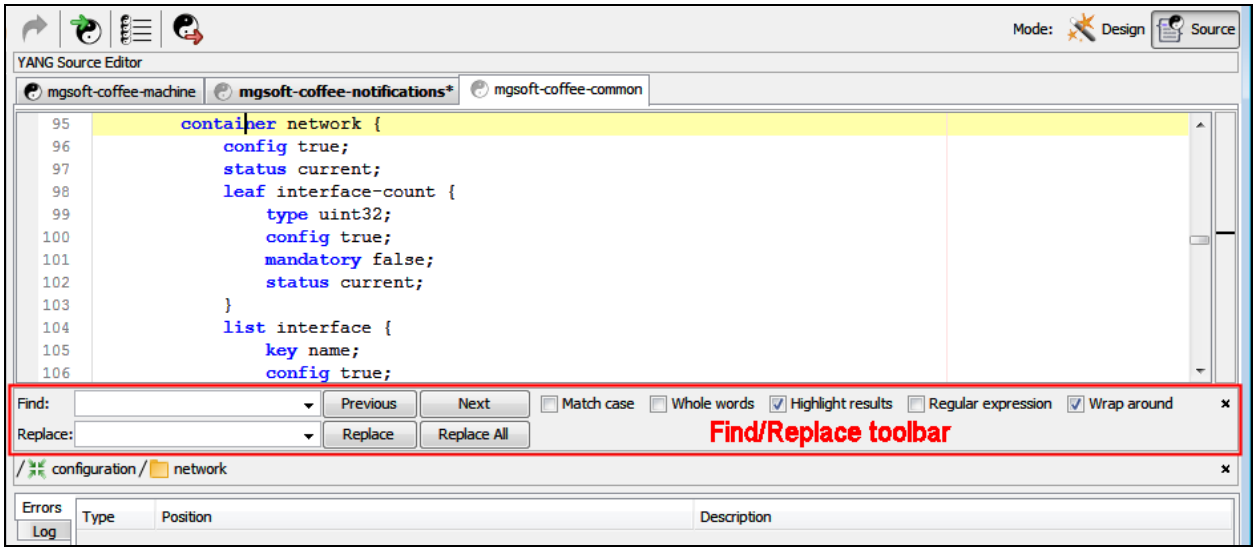


Figure 42: The Find/Replace toolbar in the Source mode

2. Into the ***Find*** input line enter one or more characters or words you are searching for. In addition, you can select the following search options in the Find/Replace toolbar:

   - Check the **Match case** checkbox to make the search case sensitive, meaning that search operation distinguishes between uppercase and lowercase letters. If this option is enabled, the search will find only those strings in which the capitalization matches the one used in the search query (e.g., `Current` will find `Current`, but not `current`).
   - Check the **Whole words** checkbox to find only those strings that are whole words and not part of a larger word (e.g., `link` will find `link Down`, but not `linkDown`).
   - Check the **Highlight results** checkbox to highlight found text in orange color. This option is enabled by default.
   - Check the **Regular expressions** checkbox to enable searching text by using the advanced search capabilities offered by the Java regular expressions (for more information about using the regular expressions, please visit the following site: http://www.regular-expressions.info/java.html).

- Check the **Wrap around** checkbox to enable wrap-around search, i.e., the program will search for the text you specify beginning at the current cursor position, and will continue past the end, to the beginning of the document back to the cursor position. If this option is disabled, the search begins at the current cursor position and ends at the end of the document or selection. This option is enabled by default.

3. The software searches the document or the selection for the text string that matches the search pattern. By default, the **matches** (all occurrences of the found text) are **highlighted** in orange color and the **annotation stripe** along the right border of the YANG Source Editor panel displays the quantity of matches and their relative locations (by line numbers) with orange rectangle marks (▭) (Figure 43). Click an orange mark in the annotation stripe to quickly jump to the line in the document, referenced by that mark.

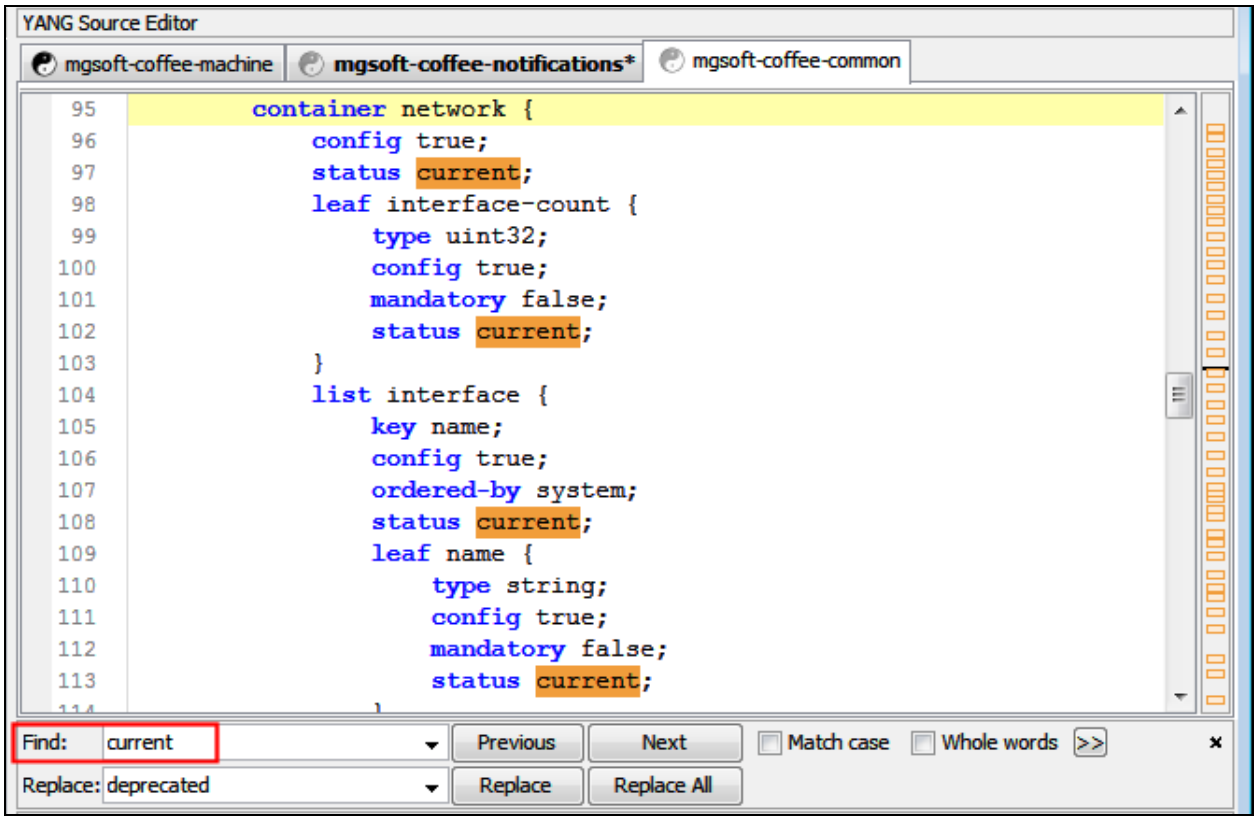


Figure 43: Finding and replacing text (matches are highlighted and displayed in annotation stripe)

4. Click the ***Next*** button in the Find/Replace toolbar to select the first occurrence of the found text.
5. Enter the replace string into the ***Replace*** input line. To simply delete the found text, leave the ***Replace*** input line empty.
6. Click the ***Replace*** button to replace the first occurrence with the replace text (e.g., replace `current` with `deprecated`).
7. Click the ***Replace All*** button if you wish to replace all occurrences of the found text with replace text.

8. Otherwise, click the ***Next*** button to find the next occurrence of the text you want to replace and then click the ***Replace*** button. Repeat this process until all desired occurrences of the found text is replaced.
9. To stop the find and replace operation, delete the search string from the ***Find*** input tine or click the ***Close*** ( ) button in the right section of the Find/Replace toolbar to hide it.

7 OPEN EXISTING YANG MODULES FOR EDITING

MG-SOFT Visual YANG Designer lets you open existing NETCONF modules in native YANG format (.yang), in XML-based YIN format (.yin), or in YANG Designer Document format (.ydd) for further editing in YANG Designer, either in Design mode or in Source mode.

7.1 Opening YANG and YIN Modules

To load a YANG or YIN module (and all dependent modules it imports and includes):

1. Copy the YANG or YIN module you want to edit to a local folder of your choice.
2. In YANG Designer, select the ***File / Open / Yang File*** or the ***File / Open / Yin File*** command from the main menu (Figure 44).

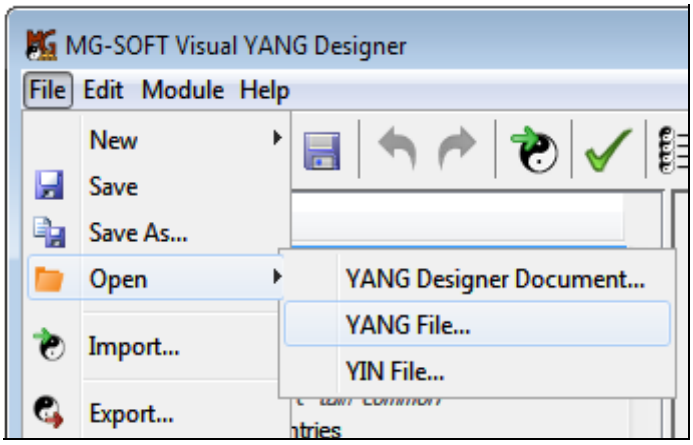


Figure 44: Selecting commands for opening existing files in YANG Designer

3. The Open YANG File / Open YIN File dialog box appears (Figure 45). In this dialog box, navigate to the folder containing the private YANG or YIN module(s), respectively, select the desired file and click the ***Open*** button.

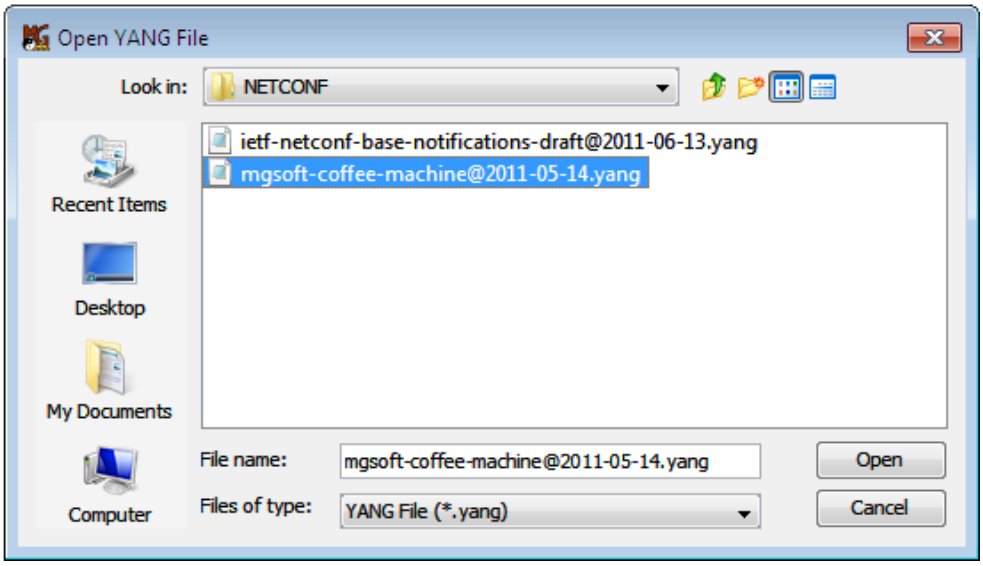


Figure 45: Selecting the YANG module to open into YANG Designer

4. YANG Designer first starts **scanning** the current folder for files that contain valid YANG and YIN modules to build a list of known modules, i.e., for all modules it detects, the module name, its revision and full path of the file defining this module is stored in the program cache (known modules list) for future use. When done, YANG Designer stars compiling the selected YANG module or parsing selected YIN module and checking its consistency. While compiling/parsing the module, the progress and error messages are being logged in the Log window panel at the bottom of the main window. During this process, the module is also checked for dependencies, i.e., the modules it imports and submodules it includes (this is done recursively for all referenced modules).

   - If any dependency is found that is not "known" to YANG Designer, the Module Load Request dialog box appears (Figure 46) prompting you to specify the location of the file that defines the referenced (sub)module.

> **Note:** "Known modules" are those modules for which information in the program cache already exists, i.e., the standard modules that ship with YANG Designer and those private modules that have already been loaded in YANG Designer.

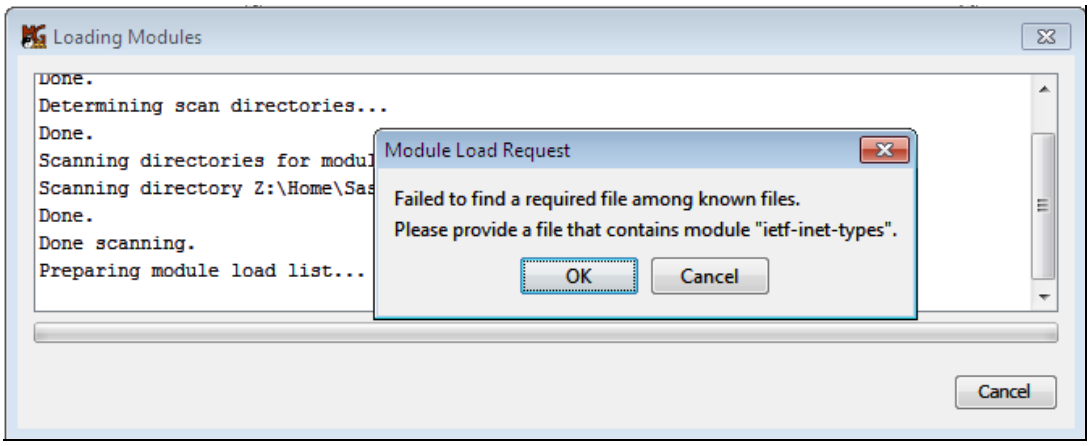


Figure 46: YANG Designer prompts you to provide the location of the required module

   - Click the ***OK*** button in the Module Load Request dialog box to close it and display the Load Module dialog box. In the Load Module dialog box, navigate to the file containing the definition of the required module, select it, and click the ***Open*** button (Figure 47).

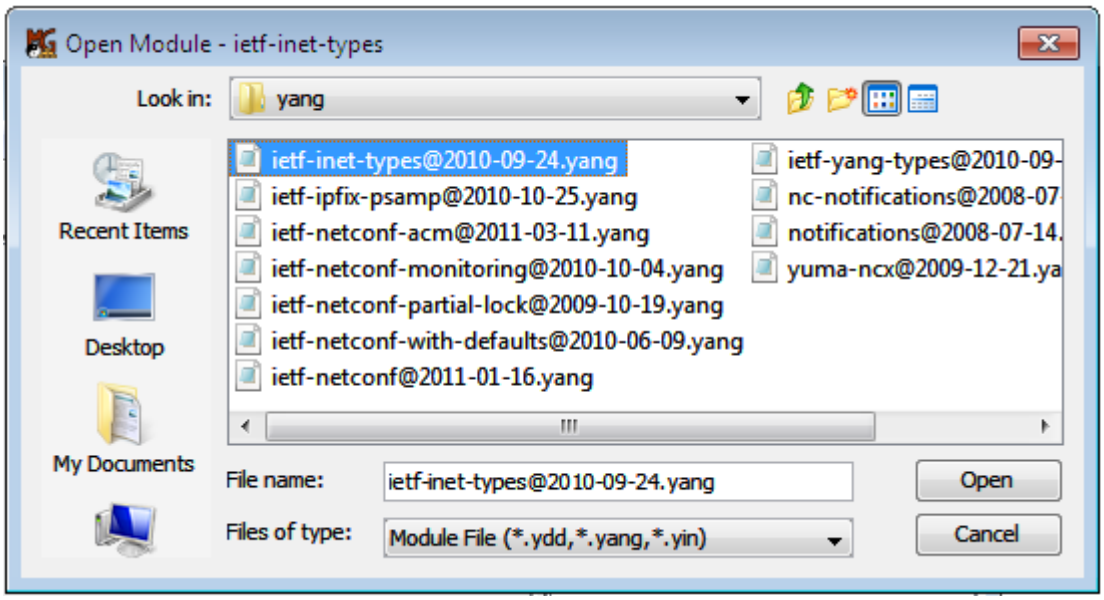


Figure 47: Specifying the location of the imported module in the Open Module dialog box

- Again, the specified folder is first scanned for files that contain valid YANG and YIN modules to build a list of known modules, then the selected module is loaded.
- If you are prompted for any other missing module, repeat the above procedure to specify its location.

Note: Once a module has been loaded from disk, this module and all other modules from the same folder (and optionally all its subfolders) are "registered" and can later be loaded from the **Known Modules** dialog box.

5. After the selected module and its dependencies (if any) have been successfully compiled (YANG) or parsed (YIN), the module and its submodules (if any) are loaded and displayed in Visual YANG Designer.
   - If using the Design mode of operation, the hierarchical structure of loaded module(s) and submodule(s) is displayed in the YANG Tree panel in the left portion of the main window (Figure 48). Imported modules, on the other hand, are loaded and displayed in the Components panel – Imports tab.
   - If using the Source mode of operation, the structure of loaded module(s) and submodule(s) is displayed in the YANG Tree Preview panel in the left portion of the main window (Figure 48), while the YANG Source Editor in center section displays the YANG code of the loaded module. Imported modules, are loaded and displayed in the Imports panel on the lower left portion of the Source mode desktop.
6. You can edit the opened module either in Design mode or in the Source mode.

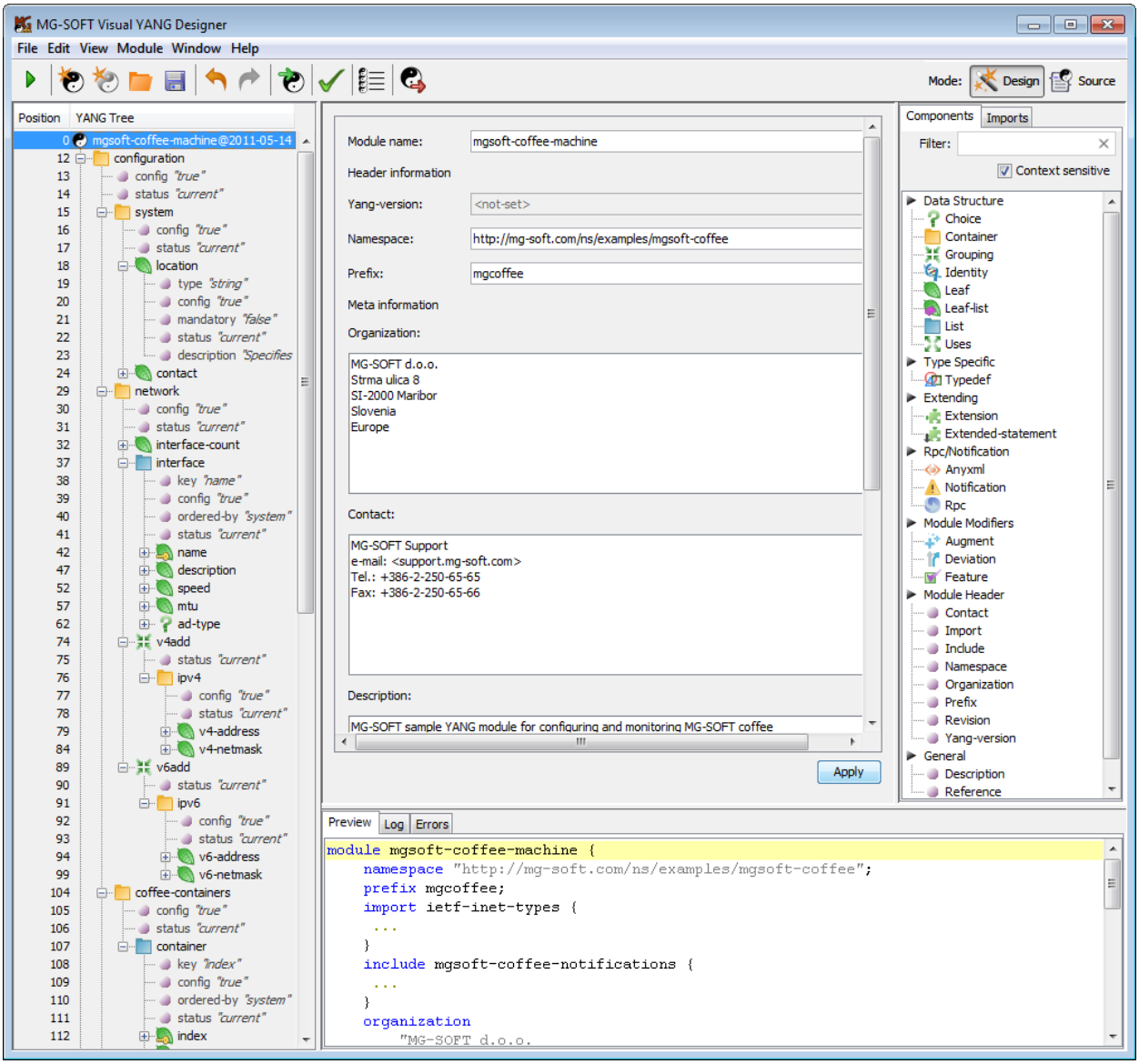


Figure 48: Opened module is displayed in YANG Designer for further editing

The location of the standard YANG and YIN modules bundled with YANG Designer depends on the operating system:

Windows: My Documents\ MGSOFTYangDesigner\modules\

Windows Vista, Windows 7, Windows Server 2008, Windows 8, Windows Server 2012:
C:\Users\[username]\Documents\MGSOFTYangDesigner\modules\yang
C:\Users\[username]\Documents\MGSOFTYangDesigner\modules\yin

Windows XP, Server 2003:
C:\Documents and Settings\[username]\My Documents\MGSOFTYangDesigner\modules\yang
C:\Documents and Settings\[username]\My Documents\MGSOFTYangDesigner\modules\yin

Linux: ~/Documents/MGSOFTYangDesigner/modules/

/home/[username]/Documents/MGSOFTYangDesigner/modules/yang/
/home/[username]/Documents/MGSOFTYangDesigner/modules/yin/

Mac OS X: ~/Documents/MGSOFTYangDesigner/modules/

/Users/[username]/Documents/MGSOFTYangDesigner/modules/yang
/Users/[username]/Documents/MGSOFTYangDesigner/modules/yin

7.2 Loading Known YANG and YIN Modules

Once a folder has been scanned for files that contain YANG or YIN modules, all module files from that folder (and optionally from all of its subfolders) are “registered” and can later be loaded from the Known Modules dialog box. The Known Modules dialog box also lets you view and configure dependencies for each module, i.e., the modules it imports and submodules it includes.

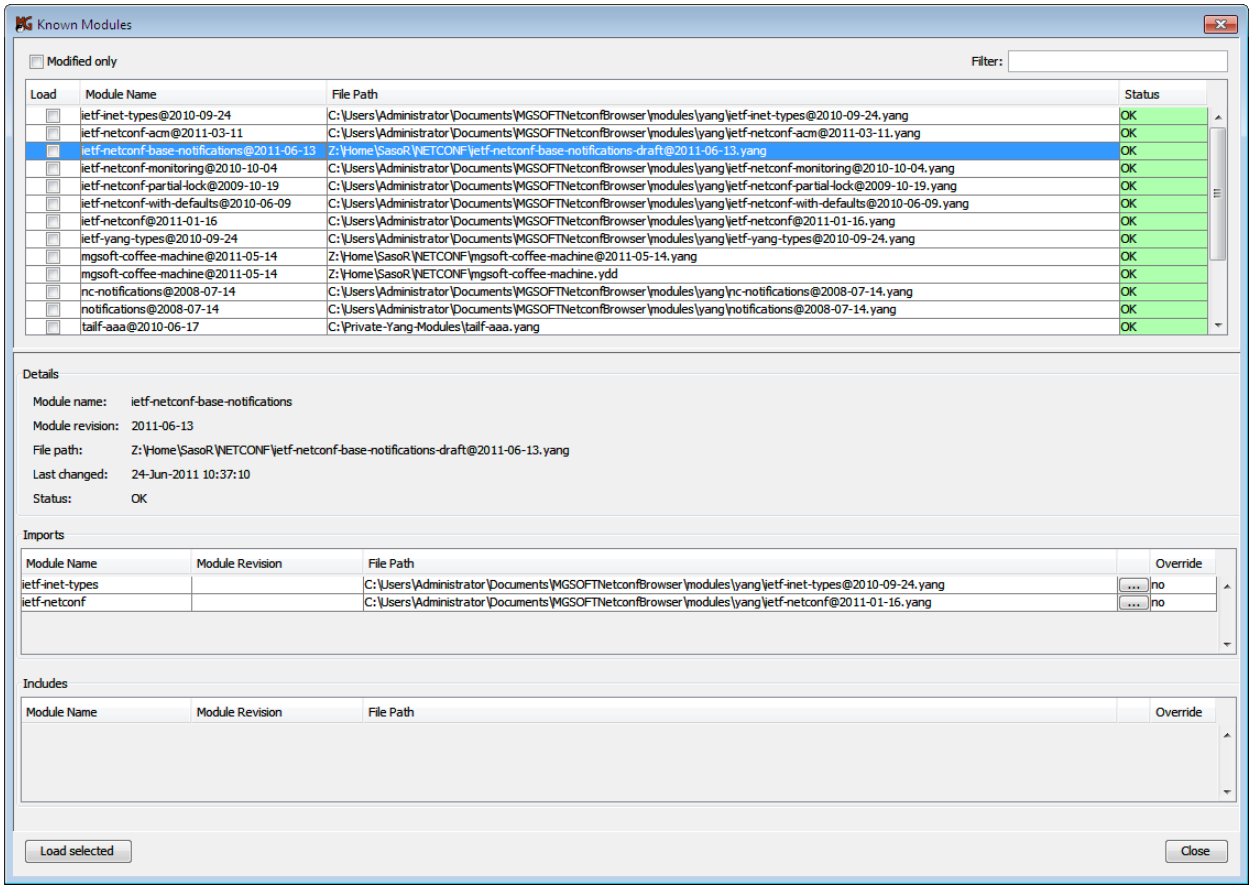


Figure 49: The Known Modules dialog box (viewing details of selected module)

To load a YANG or YIN module (and all dependent modules it imports and includes) from the Known Modules dialog box:

1. In YANG Designer, select the ***Module / Known Modules*** command from the main menu.
2. The Known Modules dialog box appears, listing all YANG and YIN modules that are currently known to YANG Designer (Figure 49).
   - Select a module in the list of known module files in the upper panel, to view the module details (location, last modified date, etc.) in the middle section, and the module files it imports and submodules files it includes in the lower panel.
   - The status column in the list of known modules indicates the **OK** status if all dependencies (imported module files and included submodule files) for the given module are available.

- If the status of a module is not **OK**, or if the status is **OK**, but you would like to change the path of the files that will be imported or included by the module, click the ***Browse*** (**…**) button next to the ***Override*** column in the list of Imports or Includes (Figure 50)

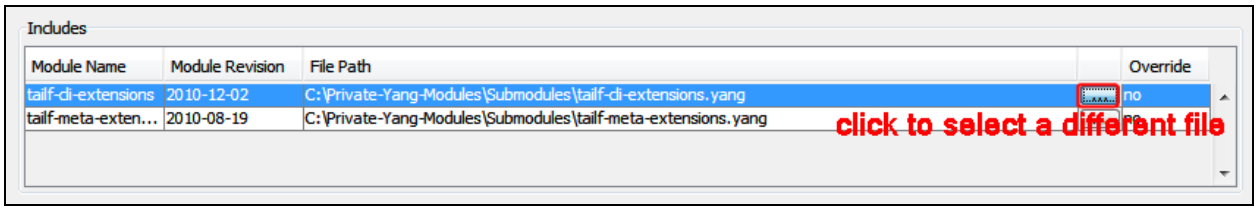


Figure 50: Specifying a different include file in the Known Modules dialog box

- In the Edit Override dialog box that appears, select the ***Browse…*** entry and click the ***OK*** button to open the Load Module dialog box (Figure 45) that lest you browse the browse the file system and select a different file containing the definition of the given submodule.

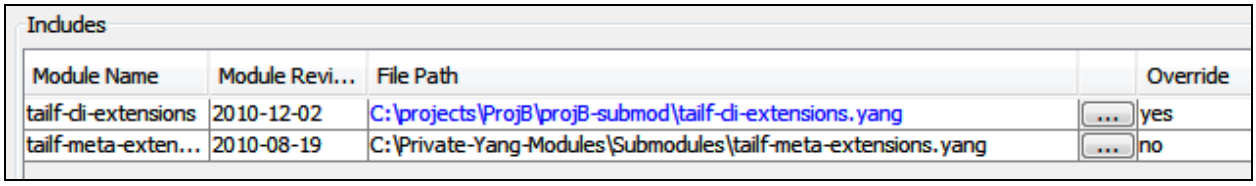


Figure 51: User overridden include submodule is displayed in blue

3. Check the Load checkbox if the module you want to load in the list of known module files and click the ***Load Selected*** button at the bottom of the Known Modules dialog box (Figure 52).

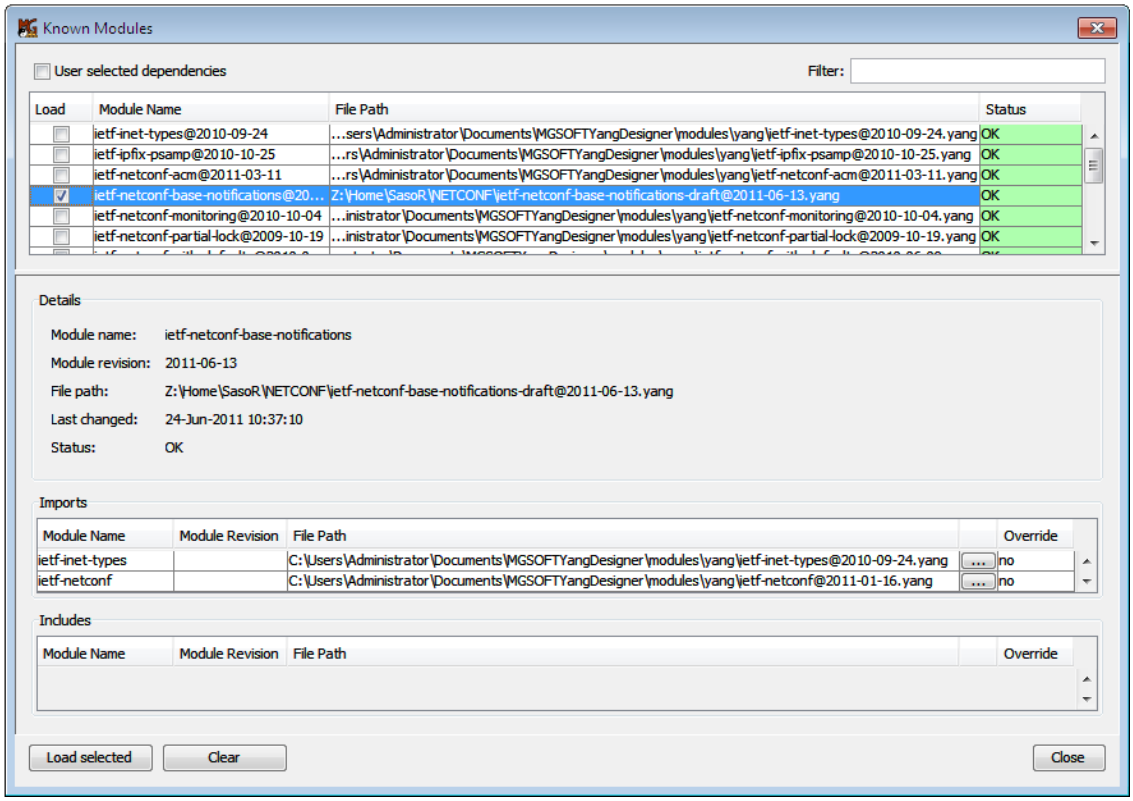


Figure 52: Loading selected modules from the Known Modules dialog box

4. Known Modules dialog box closes and the selected module and its submodules (if any) are loaded and displayed in the YANG Tree panel in the left portion of the main window where you can edit them (unless module with the same name and revision is already loaded – in such case you need to unload the currently loaded modules first and then repeat the loading procedure). Imported modules, on the other hand, are loaded and displayed in the Components panel – Imports tab.

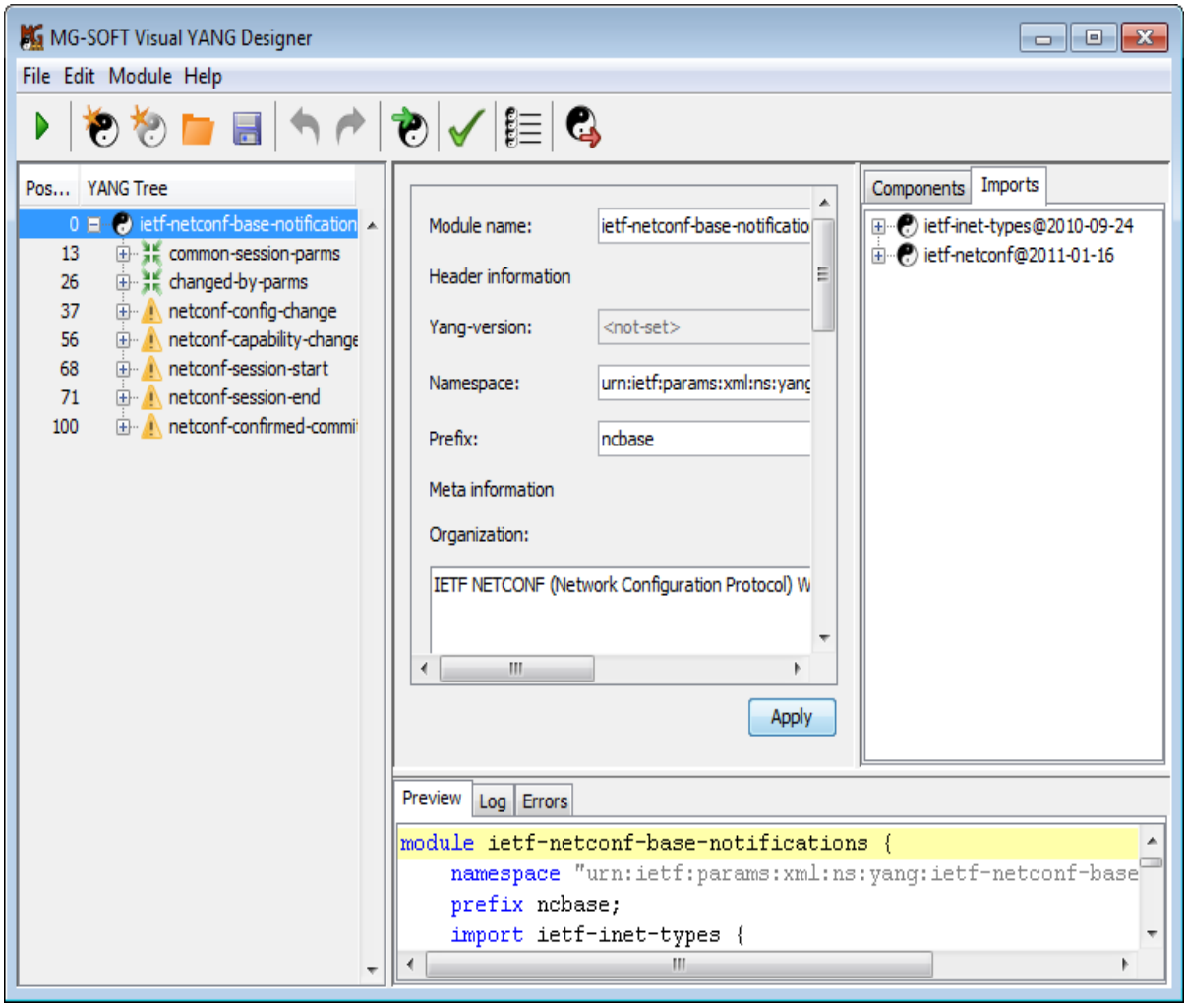


Figure 53: The Module from the Known Modules dialog box opens in YANG Designer

8 VALIDATE MODULES

MG-SOFT Visual YANG Designer lets you check whether the loaded module and its submodules (if any) comply with the YANG specification rules. This process is called module validation.

By default, Visual YANG Designer performs module validation on-the-fly, meaning that it automatically checks the validity of every change you make to the module, either in Design mode or in Source mode of operation. The software logs all detected inconsistencies in the Errors tab of the Output panel and marks all nodes or statements that do not comply with the YANG specification with either the error (⛔) or warning (⚠) overlay icon.

In Design mode, you can disable the real-time validation. To do this, open the Preferences dialog box (***Edit / Preferences***), switch to the **Design Mode** view and uncheck the ***Realtime validation*** checkbox.

In addition, the complete module validation can also be performed on the user request. This is useful when real-time validation is disabled, as described in this section.

Validation is also automatically performed before you export the module/submodule to YANG or YIN format using the ***File / Export*** command.

Visual YANG Designer uses two types of messages to report YANG module definition inconsistencies:

- An *ERROR* (⛔) *message* indicates that Visual YANG Designer has found a fatal syntactic or semantic error in the checked module definition. The reported error needs to be corrected before the resulting YANG file will conform to the YANG specification rules.
- A *WARNING* (⚠) *message* warns you of a non-fatal syntactic or semantic error in the checked YANG definitions. To generate a YANG/YIN module that conforms to the YANG rules, you should correct the inconsistencies reported by the Warning messages.

Every message contains a short, but meaningful description of the inconsistency. A list of all error and warning messages is provided in Appendix 1.

1. To check whether the current module and its submodules (if any) comply with the syntax and semantics rules of the YANG specification, use the ***Module / Validate*** command. If any inconsistencies are found during this operation, Visual YANG Designer reports them by displaying Error or Warning messages (depending on the severity level of the inconsistency) in the **Errors** tab of the Output panel displayed at the bottom of the main window (Figure 54).

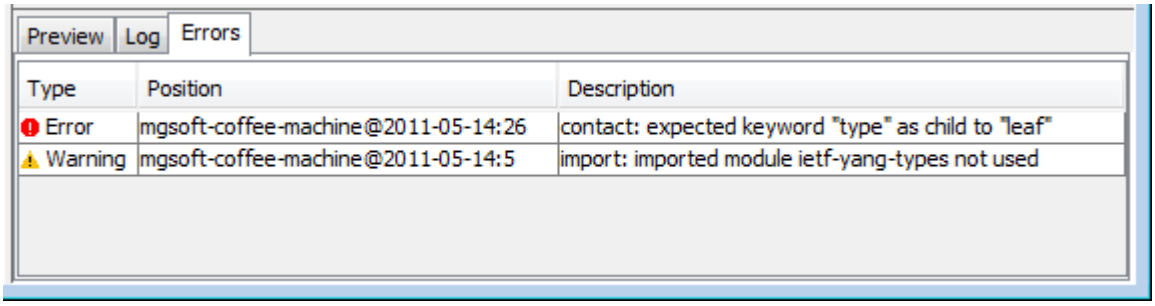


Figure 54: Validation messages displayed in the Errors tab of the Output window

2. Double-click a message in the **Errors** tab to quickly locate the:
   - node in the YANG tree (Design mode), which the message refers to, or
   - line in the YANG Source Editor (Source mode) which the message refers to.

3. To eliminate the reported inconsistency:
   - edit the properties of the referenced node in the Properties window panel (Design mode), or
   - edit the inconsistent definition in the YANG Source Editor (Source mode).

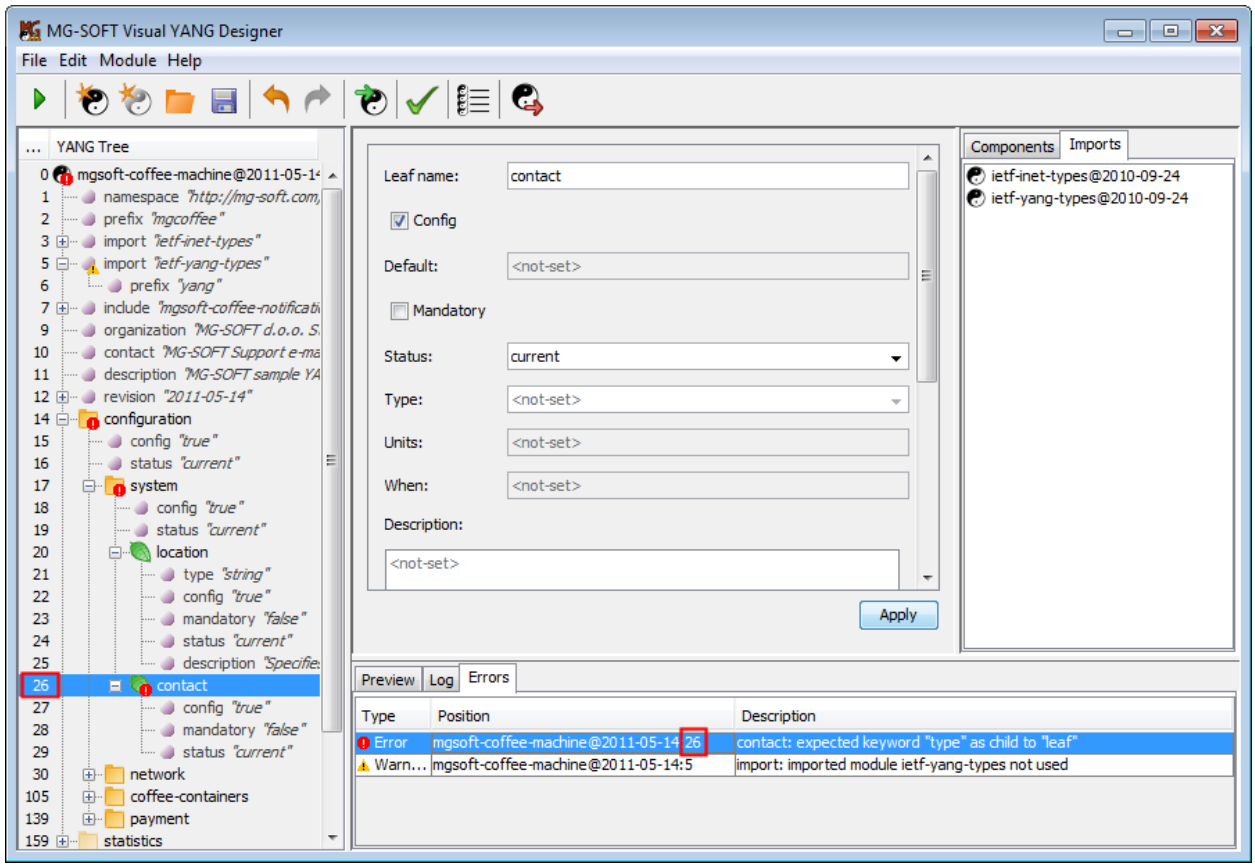


Figure 55: Double-click a message to quickly locate the corresponding node in the YANG Tree (Design mode)

9 VIEWING RESOLVED YANG TREE

The software lets you view the fully resolved YANG module tree, where the effects of **`augment`**, **`refine`** and **`deviation`** statements can be inspected in addition to seeing expanded **`uses`** statements. To view the fully resolved YANG tree, proceed as follows:

1. Select the ***Module / Show Resolved YANG Tree*** command from the main menu.
2. The **Resolved YANG Tree** window appears, displaying the tree structure of the currently selected module with resolved (expanded) YANG statement values (Figure 56).

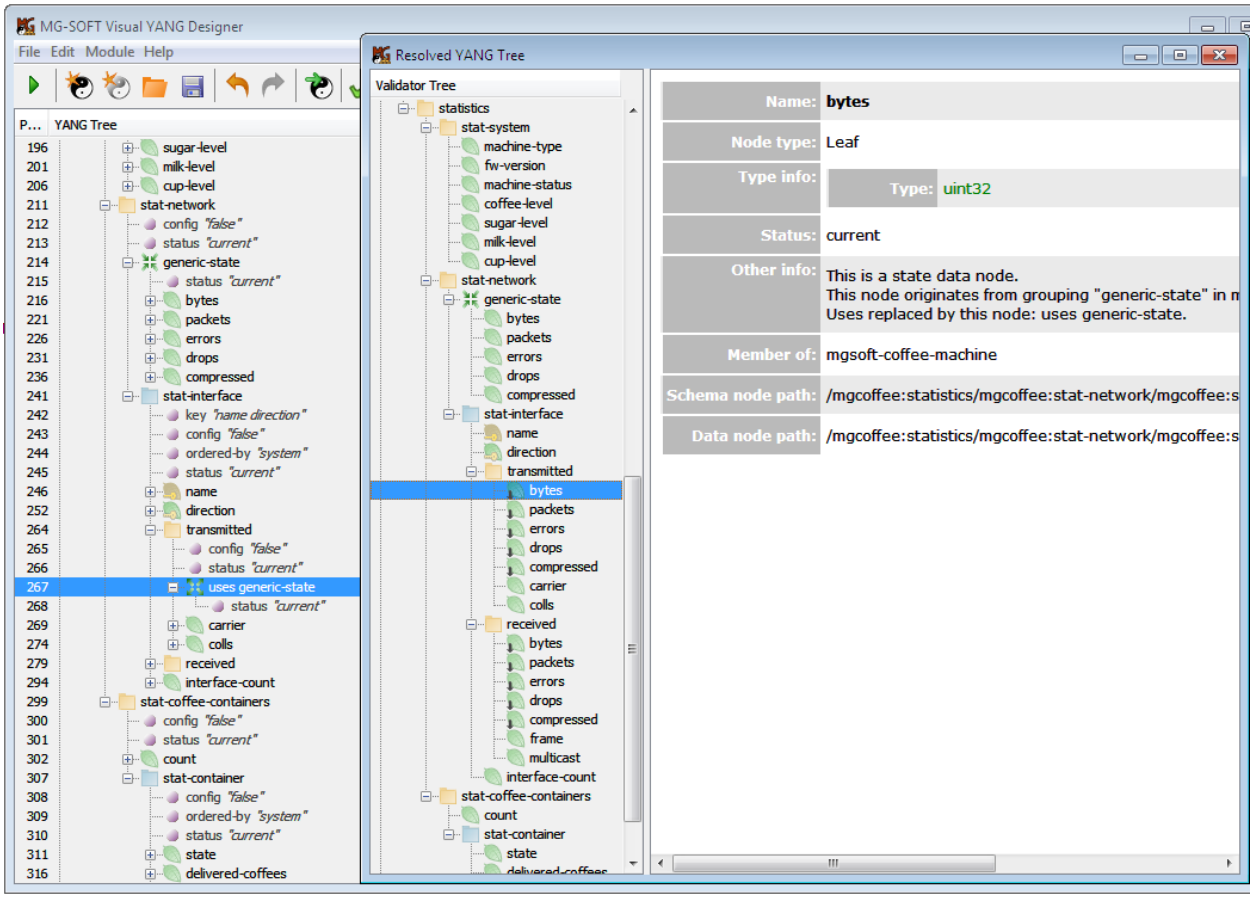

MG-SOFT Visual YANG Designer
File Edit Module Help
YANG Tree
Resolved YANG Tree
Validator Tree
Name: bytes
Node type: Leaf
Type info: Type: uint32
Status: current
Other info: This is a state data node.
This node originates from grouping "generic-state" in n
Uses replaced by this node: uses generic-state.
Member of: mgsoft-coffee-machine
Schema node path: /mgcoffee:statistics/mgcoffee:stat-network/mgcoffee:s
Data node path: /mgcoffee:statistics/mgcoffee:stat-network/mgcoffee:s

Figure 56:Viewing a module's resolved YANG tree with expanded YANG statements

3. Certain types of nodes in the Resolved YANG tree are marked with special symbols:
   - Nodes that originate from **grouping** statements are marked with the ⬇ overlay symbol,
   - nodes inserted by the **augment** statements are displayed with the ✚ symbol,
   - nodes that originate from **grouping** statements defined **in augmentatio**n modules are marked with the ⬇✚ symbol.
4. Click a node in the Resolved tree panel on the left to view its properties in the panel on right hand-side.

10 EXPORT MODULE TO YANG (.YANG) OR YIN (.YIN) FORMAT

In order to use a YANG/YIN module in any application other than MG-SOFT YANG Designer, it is necessary to export the module (and its submodules) into the YANG or YIN format. YANG and YIN modules can then be imported into and used by all NETCONF applications that support loading modules written in the native NETCONF data modeling language (YANG) or in its alternative, XML-based translated format (YIN). An example of such application is MG-SOFT NetConf Browser that can load both, YANG and YIN modules.

> **Tip:** To view YANG definition of the entire module (and its submodules), select the ***File / Export Preview*** command or switch to the ***Source mode***.

1. To export the current module and all its submodules, select the ***File / Export*** command. The Export Module dialog box appears.

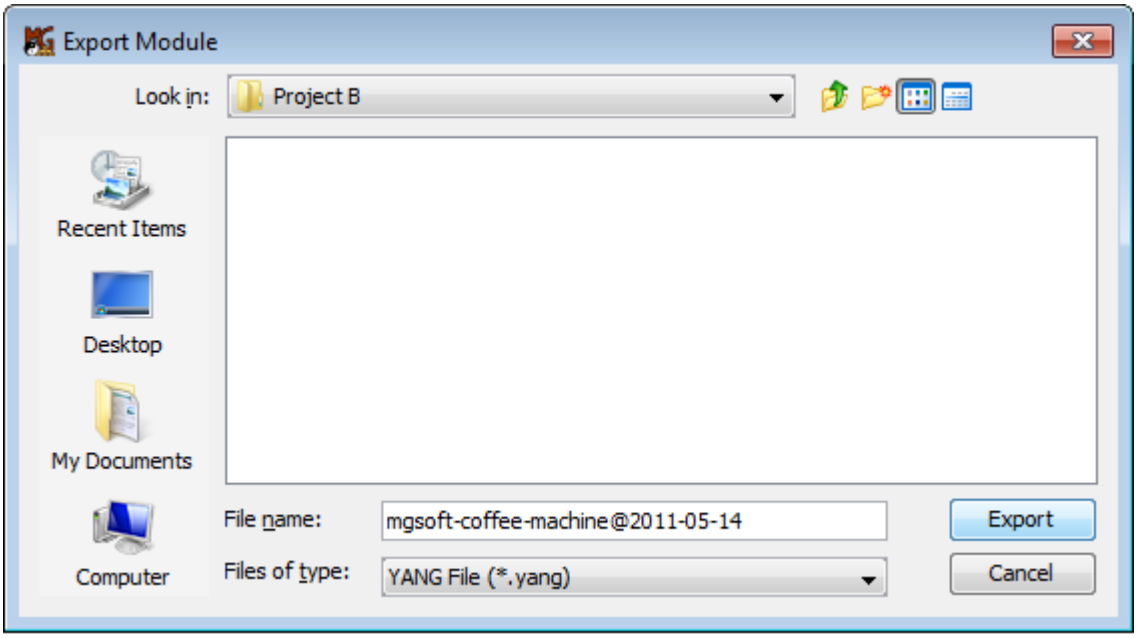


Figure 57: Exporting a finished module to YANG file format

2. In the Export Module dialog box, navigate to the destination, where you want the exported file to be created, select the **`YANG File (.yang`**) or **`YIN File (.yin)`** format from the ***Files of type*** drop down list, and enter the file name into the ***File name*** input line.
3. Click the **Export** button to close the Export Module dialog box and save the module to the selected format (YANG or YIN). The file will have the .yang or .yin file extension, depending on the selected format.
4. If the module includes any submodules, Visual YANG Designer displays the Export Submodule dialog box for every submodule in the project you are exporting (Figure 58).

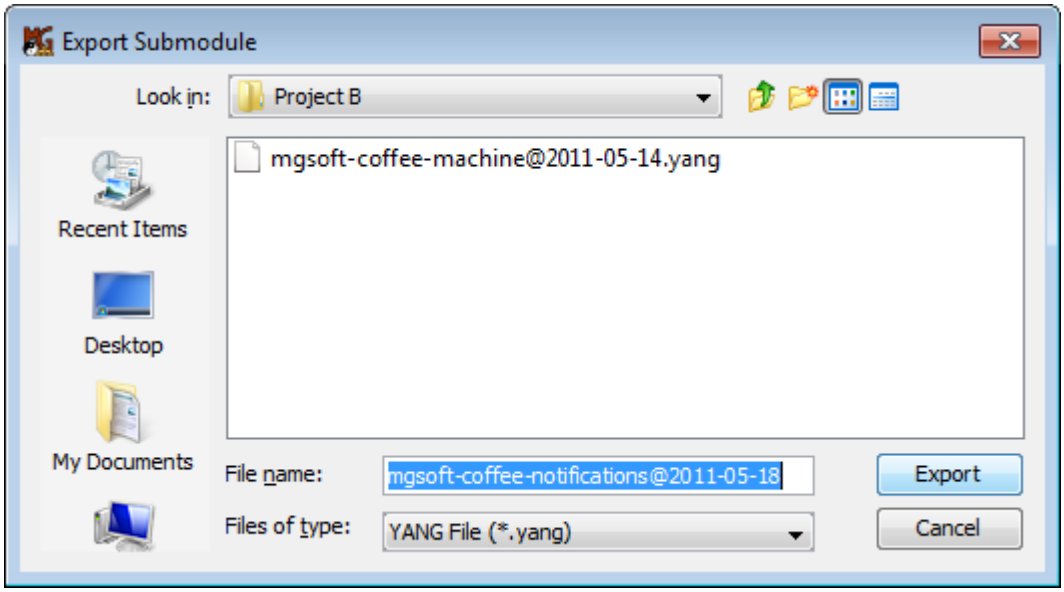


Figure 58: Exporting a submodule to a YANG format

5. Select the desired export format from the ***Files of type*** drop down list, enter the file name for the submodule into the ***File name*** input line and click the ***Export*** button to export the submodule. Repeat the procedure for all submodules in the project.

Note: Visual YANG Designer automatically validates the module and all its submodules before you export it/them the to YANG or YIN file format using the ***File / Export*** command. By default, if any inconsistencies are found, module/submodules are not exported. If you would like to export inconsistent modules/submodules, open the Preferences dialog box (***Edit / Preferences***), switch to the **General** view and check the ***Allow export with errors and warnings*** checkbox.

APPENDIX 1: LIST OF VALIDATION MESSAGES

Following is the list of **error messages** reported by MG-SOFT Visual YANG Designer:

Error Message	RFC 6020 Reference
"prefix \"%s\" already used for module %s"	7.1.4. p4
"prefix \"%s\" is not defined (reported only once)"	7.1.4. p1
"illegal type name \"%s\""	7.3. p3
"type name \"%s\" is already defined at %s"	7.3. p3
"grouping name \"%s\" is already defined at %s"	6.2.1 p1
"feature name \"%s\" is already defined at %s"	6.2.1 p1
"identity name \"%s\" is already defined at %s"	6.2.1 p1
"extension name \"%s\" is already defined at %s"	6.2.1 p1
"circular dependency for %s \"%s\""	5.1. p6 7.11. p4 7.18.1. p5 7.16.2. p4 7.3. p2
"module \"%s\" not found in search path"	5.1. p5
"module \"%s\" revision \"%s\" not found in search path"	5.1.1. p1
"module %s includes %s, but %s does not specifiy a correct belongs-to"	7.2.2. p2
"grouping \"%s\" not found in module %s"	7.12. p1
"identity \"%s\" not found in module %s"	7.16.2. p1
	9.10.2. p1
"extension \"%s\" not defined in module %s"	7.17. p3
"unexpected argument for extension %s"	7.17.2. p1
"expected argument for extension %s"	7.17.2. p1
"the value \"%s\" does not match its base type %s- %s"	7.3.2. p2 7.6.4. p2
"range error: \"%s\" is not larger than \"%s\""	9.2.4. p2
"type \"%s\" not found in module %s"	7.4. p1
"restriction %s not allowed for this base type"	9.2.3. p1 9.3.3. p1 9.4.3. p1 9.5.3. p1 9.6.3. p1 9.7.1. p1 9.8.1. p1 9.9.1. p1 9.10.1. p1 9.11.1. p1 9.12.1. p1 9.13.1. p1
"a type %s must have a %s statement"	9.3.4. p1 9.9.2. p1
"length error: \"%s\" is not larger than \"%s\""	9.4.4. p3
"length error: \"%s\" is too large"	9.4.4. p3

"a type %s must have at least one %s statement"	9.12. p2 9.6.4. p1 9.7.4. p1
"the integer value \"%s\" has already been used for the enumeration at %s"	9.6.4.2. p1
"the enumeration value \"%s\" is not a 32 bit integer"	9.6.4.2. p1
"the enum name \"%s\" has already been used for the enumeration at %s"	9.6.4. p3
"the position \"%s\" has already been used for the bit at %s"	9.7.4.2. p1
"the position value \"%s\" is not valid"	9.7.4.2. p1
"the bit name \"%s\" has already been used for the bit at %s"	9.7.4. p2
"the type %s (defined at %s) cannot be part of a union"	9.12. p3
"a \'default\' value cannot be given when \'mandatory\' is \"true\""	7.9.3. p2 7.6.4. p3
"expected a data definition statement as child to \"%s\""	7.13.2. p1 7.13.3. p1
"node %s::%s is not found"	7.12.2. p1 7.15. p1 7.18.3. p1
"node %s::%s cannot be refined"	7.12.2. p3
"\"%s\" node \"%s::%s\" cannot be refined with \"%s\""	7.12.2. p3
"the node \"%s\" is already refined at %s"	7.12.2. p3
"node %s::%s of type %s cannot be augmented"	7.15. p1
"config true cannot be set when the parent is config false"	7.19.1. p5
"there is already a child node to \"%s\" at %s with the name \"%s\" defined at %s"	6.2.1 p1
"the list at \"%s\" needs at least one key because it is used as config"	7.8.2. p1
"the list needs at least one key"	7.8.2. p1
"the key \"%s\" does not reference an existing leaf"	7.8.2. p1
"the key \"%s\" must not be listed more than once"	7.8.2. p1
"the type %s cannot be part of a key, used by leaf %s"	7.8.2. p3
"the key \"%s\" does not have same \"config\" as its list"	7.8.2. p4
"the identifier \"%s\" in the unique argument does not reference an existing container or list"	7.8.3. p1
"the unique argument \"%s\" does not reference an existing leaf"	7.8.3. p1
"the unique argument references a mix of configuration and state leafs"	7.8.3. p2
"the default case \"%s\" is not found"	9.7.3. p1
"mandatory node in default case"	9.7.3. p3
"%s:%s in the leafref path for %s at %s is not found"	9.9.2. p1
"the leafref path for %s at %s has too many \"..\""	9.9.2. p2
"%s:%s in the leafref path for %s at %s is not the name of a key leaf"	9.9.2. p3
"%s:%s in the leafref path for %s at %s is referenced more than once"	9.9.2. p3
"%s:%s in the leafref path\'s predicate for %s at %s is compared with a leaf that is not a correct leafref"	9.9.2. p2
"%s:%s in the leafref path for %s at %s has a predicate, but is not a list"	9.9.2. p3
"the leafref path for %s at %s does not refer to a leaf"	9.9.2. p1
"the leafref path for %s is config but refers to a non-config leaf %s defined at %s"	9.9.2. p4
"XPath variable \"%s\" is not defined in the XPath context"	6.4.1. p1
"XPath function \"%s\" is not defined in the XPath context"	6.4.1. p1
"XPath syntax error: %s"	6.4. p2
"key node \"%s::%s\" cannot be deviated with \"not-supported\""	7.18.3.2. p2
"the %s property already exists in node \"%s::%s\""	7.18.3.2. p3

"the %s property cannot be added"	7.18.3.2. p3
"the %s property does not exist in node \"%s::%s\""	7.18.3.2. p5
"submodule %s is included by %s, but not by the module %s"	7.1.6. p1
"syntax error in pattern: %s"	9.4.6. p1
"%s:%s in the leafref path for %s at %s references a %s node"	9.9.2. p1
"feature \"%s\" not found in module %s"	7.18.2. p1
"the type \"%s\" cannot have a default value"	9.11. p1
"cannot include %s \"%s\", must be a submodule"	7.1.6. p1
"cannot import %s \"%s\", must be a module"	7.1.5. p1
"unexpected keyword \"%s\""	12.*
"expected an argument for keyword \"%s\""	12.*
"did not expect an argument, got \"%s\""	12.*
"bad value \"%s\" (should be %s)"	12.*
"expected keyword \"%s\""	12.*
"expected keyword \"%s\" as child to \"%s\""	12.*
"unexpected keyword \"%s\", expected \"%s\""	12.*
"unexpected keyword \"%s\" with argument \"%s\", %s"	12.*
"a deviate with \"%s\" argument cannot contain %s property"	12.*

Following is the list of **warning messages** reported by MG-SOFT Visual YANG Designer:

Warning Message	**RFC 6020 Reference**
"the leaf \"%s\" occurs more than once in the unique expression"	7.8.3. p3
"default value for a key leaf is ignored"	7.8.2. p2
"\"mandatory\" statement for a key leaf is ignored"	7.8.2. p2
"explicit config statement is ignored"	7.13.2. p4 7.13.3. p4 7.14. p4
"the revision statements are not given in reverse chronological order"	7.1.9. p1
"imported module %s not used"	/
"typedef %s not used"	/
"grouping %s not used"	/
"a deviate with \"%s\" argument should contain at least one property"	12.*
"using a non-standard XPath function - %s"	6.4.1. p1

APPENDIX 2: LIST OF KEYBOARD SHORTCUTS

Action	Default Keyboard Shortcut (Windows, Linux)	Mac OS X Keyboard Shortcut	Available	Default Also Supported on Mac OS X
New Module	CTRL+N	⌘+N	Always	Yes
New Submodule	CTRL+SHIFT+N	⌘+SHIFT+N	Always	Yes
Save Module	CTRL+S	⌘+S	Always	Yes
Open Module	CTRL+O	⌘+O	Always	Yes
Undo	CTRL+Z	⌘+Z	Always	Yes
Redo	CTRL+Y	SHIFT+⌘+Z	Always	Yes
Commit Source Changes	CTRL+Enter	CTRL+Enter	Source Mode	Yes
Cut	CTRL+X	⌘+X	Always	Yes
Copy	CTRL+C	⌘+C	Always	Yes
Paste	CTRL+V	⌘+V	Always	Yes
Delete Character	Delete	Delete	Source Mode	Yes
Remove Node	Delete	Delete	Design Mode	Yes
Find	CTRL+F	⌘+F	Always	Yes
Find Next	F3	F3	Always	Yes
Find Previous	SHIFT+F3	SHIFT+F3	Always	Yes
Replace	CTRL+H	CTRL+H	Source Mode	Yes
Find Usages	CTRL+SHIFT+B	⌘+SHIFT+B	Always	Yes
Select All	CTRL+A	⌘+A	Source Mode	Yes
Format Source	ALT+SHIFT+F		Source Mode	Yes
Toggle Comment	CTRL+SHIFT+C	⌘+SHIFT+C	Source Mode	Yes
Auto-Complete Source	CTRL+Space	CTRL+Space	Source Mode	Yes
Last Edit Location	CTRL+Q	CTRL+Q	Source Mode	Yes
Go To Definition	CTRL+B	⌘+B	Always	Yes
Forward	ALT+RIGHT	ALT+RIGHT	Source Mode	Yes
Back	ALT+LEFT	ALT+LEFT	Source Mode	Yes
Reset Windows	ALT+Enter	ALT+Enter	Always	Yes
Toggle Property Nodes Off	CTRL+LEFT	CTRL+,	Design Mode	No
Toggle Property Nodes On	CTRL+RIGHT	CTRL+.	Design Mode	No
Show Property Nodes for Subtree	CTRL+SHIFT+RIGHT	CTRL+SHIFT+.	Design Mode	No
Hide Property Nodes for Subtree	CTRL+SHIFT+LEFT	CTRL+SHIFT+,	Design Mode	No
Validation Error Next	F12	⌘+F12	Always	No
Validation Error Previous	SHIFT+F12	⌘+SHIFT+F12	Always	No
Validation Warning Next	F11	⌘+F11	Always	No
Validation Warning Previous	SHIFT+F11	⌘+SHIFT+F11	Always	No